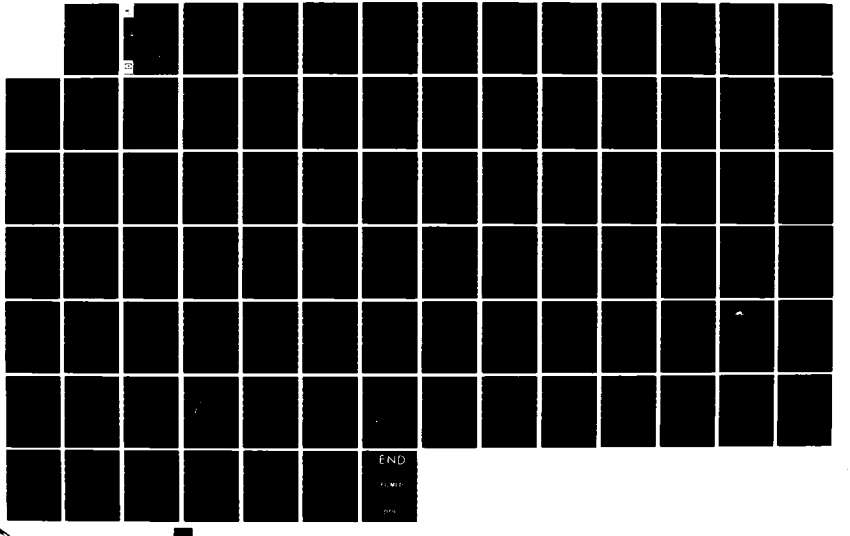
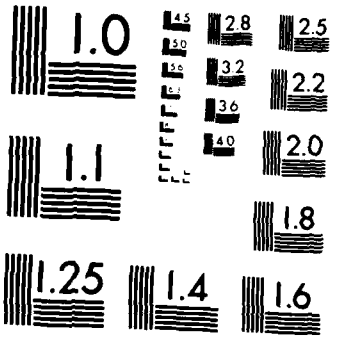
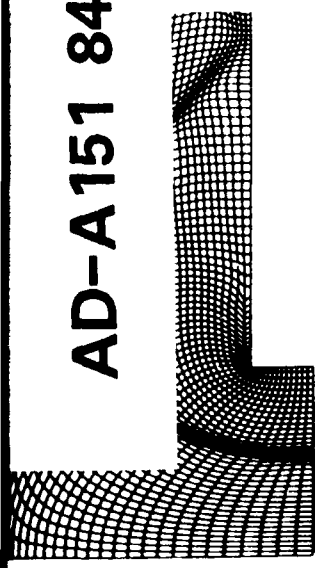
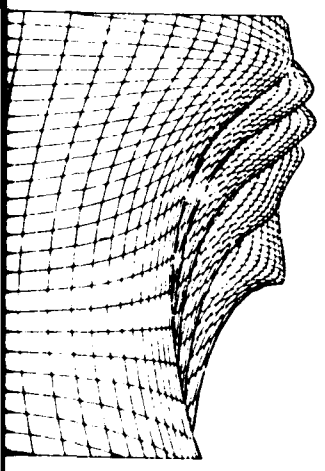
AD-A151 843 A DISCUSSION OF ADAPTIVE GRIDS AND THEIR APPLICABILITY 1/1
IN NUMERICAL HYDRO..(U) ARMY ENGINEER WATERWAYS
EXPERIMENT STATION VICKSBURG MS HYDRA..
UNCLASSIFIED B H JOHNSON ET AL. SEP 84 WES/MP/HL-84-4 F/G 12/1 NL









MISCELLANEOUS PAPER HL-84-4

(2)

A DISCUSSION OF ADAPTIVE GRIDS AND THEIR APPLICABILITY IN NUMERICAL HYDRODYNAMIC MODELING

by

Billy H. Johnson, Joe F. Thompson, A. J. Baker

Hydraulics Laboratory

DEPARTMENT OF THE ARMY
Waterways Experiment Station, Corps of Engineers
PO Box 631
Vicksburg, Mississippi 39180-0631

September 1984

Final Report

Prepared for DEPARTMENT OF THE ARMY
Assistant Secretary of the Army (R&D)
Washington, DC 20310

Under Project 4A061101A91D

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1 REPORT NUMBER: Miscellaneous Paper HL-84-4

2 GOVT ACCESSION NO.

3 RECIPIENT'S CATALOG NUMBER

4 TITLE (and Subtitle): A DISCUSSION OF ADAPTIVE GRIDS AND THEIR APPLICABILITY IN NUMERICAL HYDRODYNAMIC MODELING

5 TYPE OF REPORT & PERIOD COVERED: Final report

6 PERFORMING ORG. REPORT NUMBER

7 AUTHOR(s):
Billy H. Johnson
Joe F. Thompson
A. J. Baker

8 CONTRACT OR GRANT NUMBER(s)

9 PERFORMING ORGANIZATION NAME AND ADDRESS:
US Army Engineer Waterways Experiment Station
Hydraulics Laboratory
PO Box 631, Vicksburg, Mississippi 39180-0631

10 PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project 4A061101A91D

11 CONTROLLING OFFICE NAME AND ADDRESS:
DEPARTMENT OF THE ARMY
Assistant Secretary of the Army (R&D)
Washington, DC 20310

12 REPORT DATE: September 1984

13 NUMBER OF PAGES: 84

14 MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15 SECURITY CLASS. (of this report): Unclassified

15a DECLASSIFICATION/DOWNGRADING SCHEDULE

16 DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17 DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18 SUPPLEMENTARY NOTES

Available from National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161

19 KEY WORDS (Continue on reverse side if necessary and identify by block number)

Coordinates, Curvilinear. (LC)
Differential equations--Numerical solutions. (LC)
Error functions. (LC)
Hydrodynamics--Mathematical models. (LC)
Mathematical physics. (LC)
Numerical grid generation (Numerical analysis). (LC)

20 ABSTRACT (Continue on reverse side if necessary and identify by block number)

Due to the representation of a continuous physical domain in a discrete manner, errors normally occur in the numerical solution of partial differential equations. These errors are dependent upon the gradients of the solution variables as well as the spacing of the discrete set of points covering the domain. Historically, numerical errors have been controlled in one of two ways. Either a complex higher order solution scheme is employed or a fine

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued).

grid spacing is constructed over the entire solution domain so that low-order schemes can be used. From a programming and computational efficiency viewpoint, low-order schemes are more desirable. However, increasing the number of computational points is often an inefficient approach since portions of the domain may contain small gradients of the solution variables and a fine grid is not required there.

An approach that can sometimes be taken which reduces numerical error while still allowing for low-order schemes is the use of an adaptive grid. Such a grid moves with the developing solution so that a small grid spacing occurs in regions where the solution gradients are large with a large spacing occurring in regions where the solution is more uniform. Grid movement algorithms can be generated either by employing a variational principle or by allowing the points to attract and/or repel each other in a manner similar to the way electrical charges behave. In either case, some measure of the error is equidistributed over the grid. Although an adaptive grid normally consists of a fixed number of points, variable node grids can also be constructed by adding and/or deleting points in particular regions so as to equidistribute the numerical error.

This investigation has centered around the assessment of the usefulness of adaptive grids in the numerical solution of free surface hydrodynamics with particular problems expected to occur in the use of adaptive grids in hydrodynamic modeling being noted, e.g., the need for interpolation of the bottom topography on a moving grid. Methods of grid generation, as well as the implementation of adaptive grids, have been considered for both the finite difference and the finite element solution methods.

In general, it can be concluded that adaptive grids offer significant potential for more accurate solutions at less cost when modeling the behavior of surges, hydraulic jumps, and concentration fronts, i.e., perturbation problems containing only a few high gradient regions in the computational domain at any time. However, for problems such as the propagation of short waves in coastal regions, where many wavelengths occur in the physical domain of interest at a particular time, adaptive grids offer little or no advantage since gradients occur over essentially the complete domain.

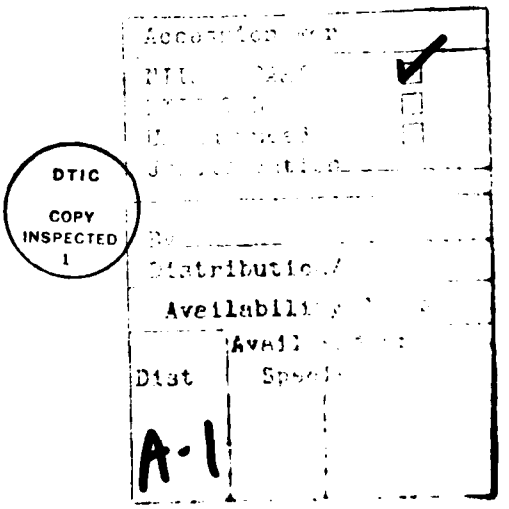
Some of the greatest potential for adaptive grid techniques in hydrodynamic modeling lies in the use of such techniques to numerically generate fixed grids in tidal circulation studies. Such studies normally involve estuaries and coastal areas containing navigation channels requiring fine grid resolution. Adaptive grid techniques employing adaption to the water depth should be of great use in developing grids for such problems, and research efforts in this direction are strongly encouraged.

PREFACE

The study reported herein was conducted during the period February 1984 to September 1984 by the Hydraulics Laboratory of the US Army Engineer Waterways Experiment Station (WES) under the general supervision of Messrs. H. B. Simmons, former Chief of the Hydraulics Laboratory, and F. A. Herrmann, Jr., Chief of the Hydraulics Laboratory, and M. B. Boyd, Chief of the Hydraulic Analysis Division (HAD). The study was funded by Department of the Army Project 4A061101A91D, "In-House Laboratory Independent Research," sponsored by the Assistant Secretary of the Army.

Dr. B. H. Johnson, HAD, directed the study and the preparation of the report, with assistance from the coauthors Dr. J. F. Thompson of the Department of Aerospace Engineering at Mississippi State University and Dr. A. J. Baker of the Department of Engineering Science and Mechanics at the University of Tennessee.

Commanders and Directors of WES during the conduct of this study and the preparation and publication of this report were COL Tilford C. Creel, CE, and COL Robert C. Lee, CE. Technical Director was Mr. F. R. Brown.



CONTENTS

CONVERSION FACTORS, US CUSTOMARY TO METRIC (SI)
UNITS OF MEASUREMENT

US customary units of measurement can be converted to metric (SI) units as follows:

Multiply	By	To Obtain
cubic feet per second	0.02831685	cubic metres per second
Fahrenheit degrees	*	Celsius degrees or Kelvins
feet	0.3048	metres
feet per second	0.3048	metres per second

* To obtain Celsius (C) temperature readings from Fahrenheit (F) readings, use the following formula: C = (5/9)(F - 32). To obtain Kelvin (K) readings, use: K = (5/9)(F - 32) + 273.15.

A DISCUSSION OF ADAPTIVE GRIDS AND THEIR APPLICABILITY IN NUMERICAL HYDRODYNAMIC MODELING

PART I: INTRODUCTION

1. When attempting to determine the hydrodynamics of water bodies such as rivers, estuaries, etc., the US Army Corps of Engineers uses one or perhaps a combination of three approaches--field investigations, physical models, and mathematical or numerical models. Field investigations may reveal what presently occurs in a water system but cannot predict what will result from changes due to new inputs to the system. Depending upon their complexity, physical models can require significant investments of capital and long construction and test periods; whereas, depending upon approximations made to the governing equations of motion and the solution technique employed, numerical models can often provide relatively low cost and highly flexible models for some problems.

2. Historically, the majority of numerical hydrodynamic models have employed the method of finite differences to solve the equations of motion on a rectangular grid with a uniform and fixed grid spacing. As a result, irregular boundaries can only be represented in a "stair-stepped" fashion. In addition, with a fixed and uniform grid spacing, numerical errors normally occur in regions where large solution gradients develop unless the grid spacing is small enough to accurately resolve the developing gradients. Such grids often contain so many computational points as to make computations uneconomical. Over the past several years, numerous efforts have been made to address these problems. The use of boundary-fitted grids in finite difference models, as well as the development of finite element hydrodynamic models, are examples of approaches for handling irregular geometry, as well as grid resolution problems, if the locations of solutions gradients are known a priori. However, the locations of these gradients are not normally known beforehand, especially when modeling moving disturbancies, e.g., surges resulting from hydropower peaking operations.

3. For the past two years, a significant amount of research has been ongoing in the aerodynamics community on the development of adaptive or moving grid techniques to address grid resolution problems. This report is

an attempt to assess the usefulness of such techniques in the numerical modeling of free surface hydrodynamic problems.

Types of Numerical/Free-Surface Hydrodynamic Models

4. Numerical hydrodynamic models can differ widely, depending upon such things as the solution technique applied to the governing differential equations representing the physical processes, the assumptions made in the derivation of the governing equations, whether the phenomena are steady or time-varying, and the spatial dimensionality considered, with perhaps the spatial dimensionality being the most commonly used delineator.

5. If the complete three-dimensional (3-D) equations of motion are integrated over a cross section, one-dimensional (1-D) models result. Such models are commonly applied in computing river hydrodynamics, e.g., the computation of floods. Averaging over either the depth or the width results in two-dimensional (2-D) models. Vertically averaged models are applicable for the computation of nearly horizontal flow in relatively shallow and well-mixed bodies of water, whereas laterally averaged models are appropriate when dealing with relatively narrow and deep bodies of water experiencing vertical stratification of the water density.

6. Even though a free surface exists on open bodies of water, some modelers have treated the surface as a rigid lid when very little motion of the free surface occurs. The surface then becomes in essence a solid boundary and the normal component of the velocity must be zero. In addition, the pressure can no longer be prescribed at the surface but rather must be computed. The pressure boundary condition then takes the form of a derivative boundary condition, i.e., a Neumman condition as opposed to a Dirichlet condition in the true free-surface case. All hydrodynamic modeling discussed in this report treats the surface as being free to move so that free-surface waves, e.g., tidal waves in estuaries, are free to be computed. In addition, only 1- and 2-D problems are considered since little research on 3-D adaptive grids has been conducted to date.

7. For completeness, a discussion of common assumptions made in free-surface hydrodynamic models, e.g., a hydrostatic pressure, as well as the governing equations for both 1- and 2-D models, are presented in Appendix A. Various types of problems to which such models are applied and suggestions on

the use of adaptive grid techniques in the application of those models are the major subjects to be discussed in the text.

Solution Techniques

8. The governing hydrodynamic equations presented in Appendix A are nonlinear partial differential equations, which in a strict mathematical sense are classified as being of the parabolic type. However, outside the boundary layer the equations exhibit a strong hyperbolic or wave character due to the dominance of the convective terms and thus are often considered as being of the hyperbolic type. In any case, because of the nonlinearity, analytical solutions do not generally exist and one must resort to numerical methods to obtain an approximation of the continuous solution of the differential equations. Such methods consist primarily of the use of either finite differences or finite elements.

Finite element method

9. In the finite element approach, the field is divided into smaller regions of convenient shapes, such as triangles or quadrilaterals, and the solution is approximated on each element by interpolation from nodal values on the element. Using a variational principle, or a weighted-residual method, the governing partial differential equations are then transformed into finite element equations governing each isolated element. These local equations are then collected to form a global system of ordinary differential (in time), or algebraic, equations including a proper accounting of boundary conditions. The nodal values of the dependent variables are then determined from solution of this matrix equation system.

10. Despite its ability to resolve complex geometry in the computational domain, the finite element method suffers from the fact that it is relatively more complicated and expensive to program. In addition, most existing hydrodynamic finite element models appear to require significantly more computational effort per time-step. However, this may be attributed to the techniques commonly employed by finite element modelers in solving the matrix equation system rather than the method itself.

Finite difference method

11. The majority of numerical hydrodynamic models, whether they be 1-, 2-, or 3-D, employ the use of finite differences to obtain solutions of the

governing equations of fluid motion. In the finite difference method, the domain of the independent variables is replaced by a finite set of points referred to as net or mesh points. One then seeks to determine approximate values for the desired solutions at these points. The values at the mesh points are required to satisfy difference equations that are usually obtained by replacing partial derivatives by partial difference quotients. The resulting set of simultaneous algebraic equations is then solved for the values of the solution at the mesh points. If the boundaries do not coincide with mesh points, the finite difference approach applied to the equations in a Cartesian coordinate system requires interpolation between mesh points to represent boundary conditions. However, through boundary-fitted coordinate transformations irregular boundaries can be accurately handled while still making use of the simplicity of finite differences to obtain solutions. The most general of such transformations, which is discussed in more detail in the next section, is a method developed by Thompson et al. (1974), which generates curvilinear coordinates as the solution of two elliptic partial differential equations.

Grids employed

12. Whether the finite element or finite difference method is employed, the body of water being modeled must be covered with a discrete set of points. These points constitute the numerical grid upon which a solution of the governing equations is sought, and depending upon the placement of these points and the relative magnitude of solution gradients, numerical errors invariably occur. As previously noted, finite element grids generally consist of a mixture of triangles and quadrilaterals. Within the framework of finite difference solutions, there have been several attempts to obtain solutions on curvilinear, as well as stretched rectangular, grids in order to better handle irregular boundaries, as well as to improve grid resolution problems and thus to decrease the numerical error.

13. Wanstrath (1977) developed a vertically averaged numerical model for computing storm surges at coastlines through the use of conformal mapping. A spatial region of prototype space is conformally mapped into a rectangle in a mathematical image plane. To provide an evenly spaced computing grid, an independent stretching of the coordinates is then performed.

14. The Waldrop and Tatom (1976) hydrodynamic model was developed to analyze hot water discharges from power plants and uses an orthogonal curvilinear grid that is generated algebraically, i.e., by interpolation. As with

the grids generated by Wanstrath through conformal mapping, these grids lack generality in that boundary point selection and grid spacing are not arbitrary. In addition, interior bodies are not allowed in either the Wanstrath or the Waldrop models. More recently, Blumberg and Herring (1983) have developed a 3-D hydrodynamic model which also handles the horizontal dimensions through the use of an orthogonal curvilinear grid.

15. In addition to Wanstrath, others such as Waldrop and Farmer (1976) and Butler (1978) have employed the use of independent stretching of the horizontal coordinates to improve grid spacing. For example, the stretching transformation in Waldrop's buoyant plume computations is

$$X = \frac{1}{c_1} \tan^{-1} \frac{x}{c_2}$$

$$Y = \frac{1}{c_3} \tan^{-1} \frac{y}{c_4} \qquad (1)$$

thus if $c_1 = \pi/2$, then at $x = \infty$, $X = 1.0$. Even increments of ΔX thus produce a close spacing of grid points near the plume entrance, where the best resolution is desired, and yet extend the region of computation far from the origin so that boundary conditions can be more easily specified. Other researchers have employed different stretching functions, e.g., Butler employs an exponential form of stretching.

16. Vertically averaged hydrodynamic models developed by Johnson (1980) and Sheng and Hirsh (1984) are examples of models employing a general nonorthogonal curvilinear grid. Through coordinate transformations, irregular boundaries and variable grid spacing can be more accurately handled while still making use of the simplicity of finite differences to obtain solutions. The extremely general coordinate transformation employed results from a method developed by Thompson et al. (1974), which generates curvilinear coordinates as the solution of two elliptic partial differential equations with Dirichlet boundary conditions, one coordinate being specified as constant on the boundaries, and a distribution of the other specified along the boundaries. No restrictions are placed on the irregularity of the boundaries, and fields containing multiple bodies or branches can be handled as easily as simple geometries. Regardless of the shape and number of bodies and regardless of the

spacing of coordinate lines, all numerical computations, both to generate the coordinate system and to subsequently solve the fluid flow equations, are done on a rectangular grid with square mesh.

17. Since the boundary-fitted coordinate system has a coordinate line coincident with all boundaries, all boundary conditions can be expressed at grid points; and normal derivatives can be represented using only finite differences between grid points on coordinate lines. No interpolation is needed, even though the coordinate system is not orthogonal at the boundary.

18. As will be discussed later in connection with generation techniques for adaptive grids, a logical choice of the elliptic generating system is Poisson's equation. Thus for a domain such as illustrated in Figure 1, the basic problem is to solve

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= P \\ \eta_{xx} + \eta_{yy} &= Q \end{aligned} \tag{2}$$

with boundary conditions

$$\begin{aligned} &\xi = \xi_1(x,y) \ , \quad \eta = \text{const, on } \Gamma_1 \ ; \quad \xi = \xi_3(x,y) \ , \quad \eta = \text{const, on } \Gamma_3 \\ &\xi = \xi_5(x,y) \ , \quad \eta = \text{const, on } \Gamma_5 \ ; \quad \xi = \xi_7(x,y) \ , \quad \eta = \text{const, on } \Gamma_7 \\ &\eta = \eta_2(x,y) \ , \quad \xi = \text{const, on } \Gamma_2 \ ; \quad \eta = \eta_4(x,y) \ , \quad \xi = \text{const, on } \Gamma_4 \\ &\eta = \eta_6(x,y) \ , \quad \xi = \text{const, on } \Gamma_6 \ ; \quad \eta = \eta_8(x,y) \ , \quad \xi = \text{const, on } \Gamma_8 \end{aligned} \tag{3}$$

The functions P and Q may be chosen to cause the coordinate lines to concentrate as desired. One form of these functions incorporated by Thompson is that of decaying exponentials.

19. Since all numerical computations are to be performed in the rectangular transformed plane, it is necessary to interchange the dependent and independent variables in Equation 2. To accomplish the transformation, the following expressions are utilized

$$f_x = \frac{1}{J}\left[\left(fy_\eta\right)_\xi - \left(fy_\xi\right)_\eta\right]$$

$$f_y = \frac{1}{J}\left[-\left(fx_\eta\right)_\xi + \left(fx_\xi\right)_\eta\right] \tag{4}$$

The system defined by Equation 2 then becomes

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} - \gamma x_{\eta\eta} + J^2(Px_\xi + Qx_\eta) = 0$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} - \gamma y_{\eta\eta} + J^2(Py_\xi + Qy_\eta) = 0 \tag{5}$$

where

$$\alpha = x_\eta^2 + y_\eta^2$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta \tag{6}$$

$$\gamma = x_\xi^2 + y_\xi^2$$

J = Jacobian of the transformation = $x_\xi y_\eta - x_\eta y_\xi$

with transformed boundary conditions from Equation 3.

20. Although the new system of equations is more complex than the original system, the boundary conditions are specified on straight boundaries and the coordinate spacing in the transformed plane is uniform. Computationally, these advantages outweigh disadvantages resulting from the extra complexity of the equations to be solved.

21. The rectangular transformed grid is set up to be the size desired for a particular problem. Since the actual values of ξ and η are irrelevant in the transformed plane, the η-lines are assumed to run from 1 to the number of η-lines desired in the physical plane. Likewise, the ξ-lines are numbered 1 to the number of ξ-lines. The grid spacing in both the ξ and η directions of the transformed plane is taken as unity since these spacings cancel out of all difference expressions.

22. The same procedure can be extended to regions that are more than doubly connected, i.e., have more than two closed boundaries, or equivalently,

more than one body within a single outer boundary. A river reach containing more than one island is an example.

23. As a final note, both conformal and orthogonal coordinate systems are special cases of boundary-fitted coordinates as generated from elliptic systems. As will be seen later, the most promising approach for generating adaptive grids, which is based upon variational principles, is a natural extension of Thompson's work discussed above.

Adaptive Grids

Basic idea

24. In its simplest form, an adaptive grid is considered to be a boundary-conforming curvilinear coordinate system which is numerically generated such that the location of the organized set of points formed by the intersections of the coordinate lines is dictated by the physics of the problem being solved. Ultimately then, the grid points must be congregated in regions of high activity, i.e., large solution gradients, and dispersed in those regions of the physical domain where small gradients of the solution exist. A good example of a 1-D adaptive grid used in the solution of the nonlinear Burger's equation is presented in Figure 2. As shown in the x-t plot of grid points, as the moving disturbance generated by the nonlinearity becomes stronger, grid points are concentrated more and more in the high gradient region. In other words, the physics of the solution being generated dictates where the grid points are located. A good discussion of adaptive grids can be found in Chapter XI of Thompson, Warsi, and Mastin (1984), and a recent survey has been given by Thompson (1984).

Need for adaptive grids

25. When using numerical techniques to compute solutions of partial differential equations, continuous problems are cast in a discretized form. As a result, numerical errors are generally always present. A simple illustration of this is provided by the commonly used implicit forward in time and centered in space finite difference (FTCS) scheme applied to the transport equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \qquad (7)$$

where

ϕ = any concentration

u = fluid velocity

x,t = spatial distance and time, respectively

Applying the implicit FTCS method yields

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = - \frac{1}{2\Delta x} \left[(\phi u)_{i+1}^{n+1} - (\phi u)_{i-1}^{n+1} \right] \tag{8}$$

Evaluating each term by expanding in a Taylor series about (x,t) and substituting into the difference equation yields, to $0(\Delta t^2, \Delta x^3)$, the differential equation

$$\frac{\partial \phi}{\partial t} = - \frac{\partial \phi u}{\partial x} + \alpha_e \frac{\partial^2 \phi}{\partial x^2} \tag{9}$$

where

$$\alpha_e = \frac{\Delta t u^2}{2} - \frac{\Delta x^2}{2} \frac{\partial u}{\partial x} \tag{10}$$

Since Equation 7 is of hyperbolic form, for accuracy the numerical domain of influence should not exceed that imposed by the differential equation, i.e.,

$$\Delta t \; u < \Delta x \tag{11}$$

Using this, the expression for α_e can be written as

$$\alpha_e \simeq \frac{1}{2} \left(\frac{\Delta x^2}{\Delta t} \right) \left(1 - \frac{\Delta x}{u} \frac{\partial u}{\partial x} \right) \tag{12}$$

Since α_e represents a diffusion coefficient, physically its value can never be less than zero. Therefore

$$\frac{\partial u}{\partial x} < \frac{u}{\Delta x} \tag{13}$$

i.e., flow gradients must be sufficiently resolved, which in turn is accomplished by reducing the grid spacing Δx. Obviously, one way to accomplish this is to increase the number of grid points. However, if significant portions of the physical domain contain relatively small solution gradients at any particular time, much of the computational effort is wasted.

26. In addition to numerical errors associated with stability problems, implicit schemes that are unconditionally stable still generally contain numerical errors as a result of the grid spacing that influence the accuracy of the solution. As an example, consider the simplified form below of the long-period water wave equations.

$$\frac{\partial \zeta}{\partial t} + h \frac{\partial u}{\partial x} = 0$$

$$\frac{\partial u}{\partial t} + g \frac{\partial \zeta}{\partial x} = 0 \tag{14}$$

where

h = total water depth

ζ = surface elevation relative to some datum

If the difference equations are written as

$$\zeta_i^{n+1} - \zeta_i^n + \frac{1}{2} \frac{\Delta t}{\Delta x} h \left(u_{i+1}^{n+1} - u_{i-1}^{n+1} \right) = 0$$

$$u_i^{n+1} - u_i^n + \frac{1}{2} g \frac{\Delta t}{\Delta x} \left(\zeta_{i+1}^{n+1} - \zeta_{i-1}^{n+1} \right) = 0 \tag{15}$$

it can be shown that the computations are unconditionally stable. However, as illustrated in Figure 3, the accuracy of both the amplitude and the phase of the computed wave is highly dependent on the number of grid points within a wavelength so that increasing the number of grid points obviously will increase the accuracy of the computations.

27. Two alternatives to increasing the number of grid points to alleviate stability and/or accuracy problems exist. Higher order solution schemes could be constructed. However, such schemes are invariably more complex and difficult to program, especially near boundaries. In addition, unless

filtering schemes, such as the flux-corrected transport method of Zalesak (1979), are employed higher order schemes can often accentuate the solution "wiggles" resulting from the cell Reynolds number being greater than 2. Instead of improving the solution, a deterioration may occur even though the solution is more accurate from an order of truncation error analysis. The second alternative is to continue using simple lower order solution schemes but to allow the grid points to move so that the grid spacing becomes smaller in regions where solution gradients are developing, thereby increasing the resolution where needed and thus decreasing numerical errors. In other words, compute the grid as part of the overall solution. The method of characteristics, as applied in the solution of hyperbolic equations, can be considered as a type of adaptive grid.

PART II: IMPLEMENTATION OF AN ADAPTIVE GRID

28. When employing a moving or adaptive grid to obtain more accurate numerical solutions of partial differential equations with a limited number of points, a major question to be resolved is how the grid is to be generated as time marches on. Various grid generation techniques will be discussed later. Irrespective of how the new grid is generated, a separate question concerns the actual implementation of the time-dependent grid within the overall solution scheme for solving the differential equations governing the physical problem.

Finite Difference Modeling

29. The use of adaptive grids in the solution of the hydrodynamic equations of motion presented in Appendix A by the finite difference (FD) method is best based upon the use of boundary-fitted curvilinear coordinates. As discussed in paragraph 19, since all numerical computations to solve the hydrodynamic equations are to be performed in a rectangular transformed plane (Figure 1), it is necessary to interchange the independent variables (x,y) in the physical space and the (ξ,η) variables in the transformed space. Equation 4 is used to accomplish such a transformation. The resulting transformed equations have the curvilinear coordinates, ξ and η, as independent variables, so all computations can be done on the rectangular transformed region regardless of the shape of the physical region. It might be noted that in addition to transforming the independent variables the components of the velocity vector can also be transformed to yield equations involving either the contravariant or covariant velocities. This approach has been taken by Sheng and Hirsh (1984) in their development of a vertically averaged flow model.

30. A simplified form of the vertically averaged hydrodynamic equations given in Appendix A, where the equations have been transformed by using Equation 4, is given below

$$\text{Continuity:}\quad \frac{\partial \phi}{\partial t} + \frac{1}{J}\left[y_\eta (hu)_\xi - y_\xi (hu)_\eta + x_\xi (hv)_\eta - x_\eta (hv)_\xi \right] = 0 \qquad (16)$$

x-Momentum: $$\frac{\partial u}{\partial t} + \frac{1}{J}(uy_\eta - vx_\eta)u_\xi + \frac{1}{J}(vx_\xi - uy_\xi)u_\eta$$

$$+ \frac{1}{J} y_\eta g\phi_\xi - \frac{1}{J} y_\xi g\phi_\eta + \frac{gu(u^2 + v^2)^{1/2}}{hC^2} = 0 \qquad (17)$$

y-Momentum: $$\frac{\partial v}{\partial t} + \frac{1}{J}(uy_\eta - vx_\eta)v_\xi + \frac{1}{J}(vx_\xi - uy_\xi)v_\eta$$

$$+ \frac{1}{J} x_\xi g\phi_\eta - \frac{1}{J} x_\eta g\phi_\xi + \frac{gu(u^2 + v^2)^{1/2}}{hC^2} = 0 \qquad (18)$$

All symbols are defined in Appendix A.

31. Assuming that the total number of grid points remains the same, an adaptive grid could be implemented with a solution of the equations as written above. The equations would first be solved on the initial grid. For example, the initial grid might have been generated using Thompson's coordinate generation code (Thompson 1983a). Information from this solution would then be used to generate a new grid satisfying certain criteria to be discussed later. In order to continue the time marching of the solution on the new grid, initial values of the solution variables are required at each grid point. These would be obtained through an interpolation of the computed solution at the current time level from the old grid. Depending upon the accuracy of the interpolation used, errors can be introduced. In any event, the interpolation will result in increased computer time. In addition, with this implementation, the generation of the grid will always lag the flow solution. Obviously, it is not necessary to recompute the grid each time-step if solution gradients aren't developing rapidly.

32. A more natural implementation of an adaptive grid in a finite difference solution involves transforming not only the spatial (x,y) derivatives but also the time derivative. With this approach, no interpolation is required. As in the derivation of the transformation relations given by Equation 4, it can be shown that the time derivative in the (x,y) system can be transformed as below

$$\left(\frac{\partial f}{\partial t}\right)_{x,y} = \left(\frac{\partial f}{\partial t}\right)_{\xi,\eta} - \frac{1}{J}\left(f_\xi y_\eta - f_\eta y_\xi\right)\left(\frac{\partial x}{\partial t}\right)_{\xi,\eta} + \frac{1}{J}\left(f_\xi x_\eta - f_\eta x_\xi\right)\left(\frac{\partial y}{\partial t}\right)_{\xi,\eta} \qquad (19)$$

with f as an arbitrary variable. Note that while the time derivative on the left-hand side is taken at a fixed point (x,y) in physical space, the time derivative on the right is at a fixed point (ξ,η) in the transformed space, i.e., at a given grid point rather than at a given physical location. Equations 16-18 would each contain two additional terms reflecting the influence of the movement of the grid points. Thus the implementation of an adaptive grid in the hydrodynamic solution would merely involve the inclusion of the additional terms and the computation of $\partial x/\partial t$ and $\partial y/\partial t$ at each grid point in the transformed rectangular plane. These derivatives can be viewed as grid speeds and are computed from the new locations of the grid points obtained from the adaptive grid generator.

33. By implementing an adaptive grid through the transformation of the time derivative, the grid and the flow solution can be implicitly coupled. This would be accomplished by selecting a grid generation technique that employs a set of differential equations for the (x,y) locations of the grid points which are dependent upon solution variables, e.g., the water-surface elevation. These differential equations could then be solved simultaneously with the governing flow equations to yield a solution in which the grid and the flow are computed at the same time level. Obviously, an explicit implementation could also be accomplished by using information about the flow solution from the old time to compute a new grid. This approach would have to be taken if the grid is not recomputed at each time-step. As Dwyer, Smooke, and Kee (1982) note, as the number of dependent variables increases and/or the problem becomes more nonlinear, the implicit implementation of adaptive grids becomes less practical than an explicit implementation.

Finite Element Modeling

A finite element algorithm

34. Consider, for example, the depth-averaged 2-D equations for free-surface hydrodynamic flows presented in Appendix A. A finite element (FE) algorithm for these equations formally states the requirement to establish an approximation $q^h(\cdot)$ to the solution set $q(x,t) \in \{\phi(\cdot), u_i(\cdot)\}$,

to be constructed on a discretization Ω^h of the solution domain $\Omega \subset R^2 \times t \in \{x_i : 1 \leq i \leq 2,\ t \geq t_o\}$. This approximation is constructed as the union of elemental contributions $q^e(x_i,t)$, which in turn are defined as expansions in the k^{th} degree polynomial cardinal basis set $\{N_k(\cdot)\}$ and time-dependent nodal expansion coefficients $\{Q(t)\}_e$, cf. Baker (1983, Ch 5). Mathematically, this statement of approximation is

$$q^h \equiv \bigcup_e q^e(x_i,t) \tag{20}$$

where

$$q^e \equiv \{N_k(x_i)\}^T \{Q(t)\}_e \tag{21}$$

35. Since Equations 20-21 define the approximation, substitution into the flow equations defines the associated error. The second step of an FE algorithm is therefore a formal statement of constraint on this error. The form of this constraint which enjoys an optimal error estimate for a linear elliptic problem statement is the so-called Galerkin statement, which requires the distribution of the error to be orthogonal to the function space $\{N_k(\cdot)\}$ used to construct q^h. For the more general problem statement wherein non-smooth solutions to hyperbolic equations are sought, the Galerkin statement is typically augmented with an additional dissipative penalty constraint, cf. Baker (1983, Ch 4). The mathematical statement of error constraint is thus of the form

$$\int_{R^2} \{N_k(\cdot)\}\, L(q^h) + \beta \cdot \int_{R^2} \nabla\{N_k(\cdot)\} L^c(q^h) \equiv \{0\} \tag{22}$$

where β is a parameter set that can be optimized.

36. Upon evaluation of the integrals in Equation 22, there results an ordinary differential equation system written on the time evolution of the expansion coefficient set in the form

$$[M] \frac{d}{dt} \{Q\} + \left(\{U\}^T [C] + \{G\} + \{G\} \right) \equiv \{0\} \tag{23}$$

where

$[M]$ = mass matrix

$\{U\}$ = convection velocity field

$[C]$ = associated (convective derivative) influence on the dependent variable set $\{Q\}$

$\{G\}$ = remaining source terms

For the time-accurate solution, Equation 23 is employed to evaluate the Taylor series

$$\{F\} = \{Q\}_{j+1} - \{Q\}_j - \Delta t \frac{d}{dt}\{Q\}_{j+\theta} - \ldots = \{0\} \tag{24}$$

Equation 24 is a nonlinear algebraic equation system for $\theta > 0$, the solution statement for which is cast using a Newton iteration algorithm in the form

$$[J]\{\Delta Q\}^{p+1}_{j+1} = -\{F\}^{p}_{j+1} \tag{25}$$

where p is the iteration index. The solution field is defined as

$$\{Q\}^{p+1}_{j+1} = \{Q\}^{p}_{j+1} + \{\Delta Q\}^{p+1}_{j+1} \tag{26}$$

and the Newton Jacobian is constructed as

$$[J] = \frac{\partial\{F\}}{\partial\{Q\}} \tag{27}$$

37. All computational models of unsteady physical conservation laws eventually produce the algebraic equation system (Equation 24) the numerical solution of which (Equations 25-27) constitutes the heart of a code. Defining an explicit time integration procedure, $\theta = 0$ in Equation 24, this equation system can be linearized, and the Jacobian (Equation 27) replaced by the identity matrix, making the linear algebra solution (Equation 25) a trivial operation. The "pain" of this choice is the restriction to use of very small integration time-steps Δt (to bound numerical evolution error). Hence, many time-steps (passes through the code) are required to predict long-time flow phenomena.

38. Conversely, definition of an implicit integration procedure ($\theta > 0$) significantly relaxes the time-step/size constraint at the expense of the

(exact) Jacobian (Equation 27) becoming a full nonlinear matrix. Hence, in comparison, forming and solving Equations 24-27 is a much more computationally intensive operation, although it need be done fewer times since Δt can be much larger. Between an explicit procedure, and use of the exact Jacobian with an implicit procedure, lies the entire family of approximate Jacobian methods, e.g., SOR, SLOR, Picard, ADI, Checkerboard, Zebra, approximate factorization (AF), tensor product (TP), etc. By choosing simple approximations (block tridiagonal), the degraded (Newton) convergence rate is compensated by code operations that are significantly less computer-resource intensive, and hence is (usually) more efficient than either explicit or exact-implicit methods. Finite difference code developers are "expert" at implementing approximate linear algebra solution techniques.

39. Upon substitution of Equation 23, Equation 24 takes the specific form

$$\{F\} = [M]\left(\{Q\}^{P}_{j+1} - \{Q\}_{j}\right) + \theta\Delta t\left(\{U\}^{T}[C]\{Q\} + \ldots\right)^{P}_{j+1}$$

$$+ (1-\theta)\Delta t\left(\{U\}\ C\ \{Q\} + \ldots\right)_{j} = \{0\} \qquad (28)$$

The extension of the finite element formulational statement, Equations 20-22, to an adaptive mesh framework requires only a modest extension in the definition of the basis $\{N_k(\cdot)\}$ upon which the approximation is cast. The argument $(\cdot)$ of the basis involves polynomials in the x_i coordinate system spanning R^2 multiplied by combinations of the coordinates X of the nodes of the elemental domain. For example, in one dimension, the simplest element domain R^1_e is the line segment with a node at the left and right ends. Denoting the corresponding values of X as X_L and X_R, and realizing that the length of the element is $\Delta_e = X_R - X_L$, the $k=1$ basis on R^1_e takes the specific form

$$\{N_I(x,X)\} = \begin{Bmatrix} 1 - \frac{x}{\Delta_e} \\ \\ \frac{x}{\Delta_e} \end{Bmatrix} \qquad (29a)$$

$$\Delta_e = X_R - X_L \qquad (29b)$$

40. In the extension to an adaptive grid, the coordinates of the nodes of the mesh are permitted (required) to move (in time), hence $X \rightarrow X(t)$. Therefore Equation 21 is generalized to the form

$$q^e(x,t) \quad \{N_k(x,X(t))\}^T \{Q(t)\}_e \tag{30}$$

Substitution of Equations 20 and 30 into Equation 22 and using the chain rule for the time derivative yields

$$\frac{dq^h}{dt} = \sum_e \frac{dq^e}{dt} = \sum_e \left[\{N_k(\cdot)\}^T \frac{d}{dt} \{Q\}_e + \frac{\partial}{\partial t} \{N_k\}^T \{Q\}_e \right]$$

$$= \sum_e \left[\{N_k\}^T \frac{d}{dt} \{Q\}_e + \{\dot{X}_e^T\} \left[\bar{N}_k\right] \{Q\}_e \right] \tag{31}$$

where $\left[N_k(x)\right]$ is a matrix with elements a function of the x coordinate system. For example, for Equation 29

$$\frac{\partial}{\partial t} \{N_1(x,X)\} = \begin{Bmatrix} \frac{x}{\Delta x_e} (\dot{X}_R - \ddot{X}_L) \\ \\ -\frac{x}{\Delta x_e} (\dot{X}_R - \dot{X}_L) \end{Bmatrix}$$

$$= \frac{1}{\Delta x_e} \{\dot{X}\}_e^T \begin{bmatrix} -\frac{x}{\Delta x_e} & \frac{x}{\Delta x_e} \\ \\ \frac{x}{\Delta x_e} - 1 & -\frac{x}{\Delta x_e} \end{bmatrix} \tag{32}$$

where the elements of $\{\dot{X}\}_e$ express the (Lagrangian) velocity of the nodal coordinates of the adaptive grid. Upon substitution of Equation 31 into the basic finite element constraint statement, Equation 22, and proceeding through

the formulational details, for implementation of an adaptive grid Equation 28 becomes modified to the form

$$\{F\} = [M]\left(\{Q\}^{p}_{j+1} - \{Q\}_{j}\right)$$

$$+ (\alpha, 1-\alpha)\Delta t\left(\{U\}^{T}[C]\{Q\} + \{\dot{X}\}^{T}[C_G]\{Q\} + \ldots\right)_{j+1,j} = \{0\} \qquad (33)$$

Thus the motion of the adaptive grid is accounted for by the grid velocity action on the solution variable $\{Q\}$ through the grid convection matrix $[C_G]$.

41. The step to use of a boundary-fitted grid (BFG) transformation, as described in detail in Baker (1983, Ch 5,8), involves transformation of the divergence operator into scalar components parallel to the coordinate lines of the BFG system. The associated scalar product with the convection velocity vector u_i produces scalar components that are also parallel to the BFG coordinate lines. Denoting this system as η_i, and proceeding through the details produces Equation 33 in the modified form

$$\{F\} = [M]\left(\{Q\}^{p}_{j+1} - \{Q\}_{j}\right)$$

$$+ (\alpha, 1-\alpha)\Delta t\left(\{UK\}^{T}[CK]\{Q\}\right.$$

$$\left. + \{\dot{X}K\}^{T}[C_G K]\{Q\} + \ldots\right)_{j+1,j} = \{0\} \qquad (34)$$

In Equation 34, the discrete index K signifies that all derivatives and convection velocity components are computed in scalar components parallel to η_i (rather than x_i).

Computational aspects

42. Implementation of the finite element algorithm statement, Equations 20-22 and 25-27, using the adaptive grid generalization and the BFG transformation provides for several computational aspects of importance. The inclusion of the grid convection contribution, $[C_G]$ in Equations 33-34, permits definition and use of adaptive methodologies to cluster refined grids

about moving fronts. A key implementation aspect relates to code efficiency. Finite element fluid mechanics codes have gained a reputation for being less efficient than their finite difference counterparts. As noted earlier, this is not a consequence of the finite element theory itself, but instead reflects directly on implementation choices for the linear algebra statement, Equations 25-27 for an implicit construction, $\theta \neq 0$ in Equation 24, as discussed. Based on the finite element structural mechanics historical development of direct, out-of-core solvers, many (most) finite element code implementations have relied on direct extension of these solver software packages to the fluid mechanics problem classes. In distinction, a direct solution (of even a linear Poisson equation) is never attempted in an FD code, but instead a matrix iteration procedure is defined that uses easily formed (usually block-tridiagonal) approximations to the Newton algorithm Jacobian (Equation 27). The computer storage requirement for the approximate Jacobian is negligible, in comparison, as is the CPU needed to execute an LU decomposition and back substitution. The computational penalty is a much reduced convergence rate for Equation 25, but each iteration proceeds so rapidly that the many grid sweeps required for the FD code are typically completed with minimal expenditure of computer resources.

43. The generalized (BFG) coordinates form of the finite element algorithm, Equation 34, provides the theoretical framework for implementation of efficient solving techniques for comparison to FD procedures. In particular, as developed in detail by Baker (1983, Ch 5) for the BFG form of the algorithm statement, using the tensor product cardinal basis $\{N_k(\cdot)\}$ for Equation 21, hence 34, facilitates reexpression of the Newton algorithm Jacobian $[J]$, Equation 27, in the outer (tensor) matrix product form

$$[J] = [J_1] \odot [J_2] \tag{35}$$

Then, for example, for the linear basis (k=1) and the governing vertically averaged hydrodynamic equations, each tensor product factor $[J_\eta]$ in Equation 35 is a three-block tridiagonal matrix, the storage requirements for which are negligible in comparison to that of $[J]$, as is the CPU execution time for solving.

44. Insertion of Equation [illegible] yields for Equation 25

$$\left[J_1\right] \otimes \left[J_2\right] \{\delta Q\}_{j+1}^{p+1} = - \{F\}_{j+1}^{p} \tag{36}$$

which is replaced in the code by the approximation

$$\left[J_1\right] \{P\}_{j+1}^{p+1} = - \{F\}_{j+1}^{p} \tag{37a}$$

$$\left[J_2\right] \{\delta Q\}_{j+1}^{p+1} \equiv \{P\}_{j+1}^{p+1} \tag{37b}$$

Equation 37a is solved by sweeping the grid lines parallel to the η_1 (BFG) coordinate to obtain the intermediate vector $\{P\}$, the entries in which are then realigned parallel with the η_2 coordinate for subsequent solving of Equation 37b. This second solution sweep yields the tensor product approximation to $\{\delta Q\}_{j+1}^{p+1}$, which is then used in Equation 26 to compute $\{Q\}_{j+1}^{p+1}$. The Newton algorithm quadratic convergence rate is compromised, of course, by using the approximation defined by Equation 37, but it is (usually) more than compensated by the much reduced computer resource requirements.

45. The definition and use of the tensor matrix product approximate Jacobian exert another influence of great importance. With elimination of the Newton Jacobian as the dominating computer (hardware) impediment, a finite element code can then afford to use a much finer discretization of the hydraulics problem domain along with an adaptive grid. The net result would be a significant improvement in solution accuracy with enhanced definition of sharp field gradients. This in turn would most likely decrease sharply the need for excessive eddy (or artificial) diffusivity, yielding an overall substantial accuracy improvement.

PART III: GENERATION OF ADAPTIVE GRIDS

46. The generation of time-dependent grids to better resolve developing flow gradients can be accomplished while either maintaining a fixed number of grid points or adding grid points in locations where some algorithm that senses local error indicates they are needed. Most of the research over the past few years on the use of adaptive grids has adopted the fixed number of points approach. For completeness, a brief discussion of the variable number of points approach is also given.

47. In order to construct a suitable grid to be used for solving a particular problem, some knowledge of the solution is required initially if the grid is to remain fixed in time. For example, a high grid-point density is desired in high gradient regions of the solution variables. However, in general, too little is known about the solution to be computed; and the locations of such high gradient regions are unknown. Therefore the establishment of a suitable grid a priori is difficult.

Fixed Number of Grid Points

48. Accepting that the best grid for a given problem depends on the solution, a variety of papers concerned with moving the grid points as the solution changes have been published by researchers in the aerodynamics field. A basic review paper by Thompson (1984) presents an extensive discussion of these publications. In addition, basic concepts of grid generation techniques are presented by Thompson (1983c). The purpose of the present discussion is not to duplicate Thompson's review but rather to focus on the two basic approaches for generating adaptive grids.

Variational approach

49. A general feature of all adaptive grid generation techniques is the minimization of numerical error by the equidistribution of some solution quantity related to the error over the grid. From a continuous viewpoint, since some quantity is to be minimized by the movement of the grid, a variational approach appears logical. As will be shown later, the grid generation approach developed by Thompson et al. (1974), is actually a special case of grid generation based upon a variational principle.

50. Numerical error. Before discussing some specifics of grid generation, a discussion of solution errors due to the grid is required. As demonstrated by Mastin (1982), and Thompson and Mastin (1983), a curvilinear coordinate system can have a significant effect, favorable or adverse, on the error in the numerical solution. This is because the truncation error is dependent not only on the higher order derivatives of the solution and the local grid spacing but also on the rate of change of the grid spacing and on the skewness of the grid, i.e., the departure of the grid lines from orthogonality.

51. In Mastin's analysis it is shown that using standard central differences to approximate first derivatives in the computational plane results in additional truncation errors, such as that given below for f_{ξ}, due to the nonuniformity and nonorthogonality of the grid.

$$T = O(h^2) - 1/2\ x_{\xi}x_{\xi\xi}f_{xx} - 1/2\ (x_{\xi}y_{\xi\xi} + y_{\xi}x_{\xi\xi})\ f_{xy}$$

$$- 1/2\ y_{\xi}y_{\xi\xi}f_{yy} + H.\ O.\ T. \qquad (38)$$

where

h = the grid spacing

f = an arbitrary variable

Considering a rectangular grid, if the grid spacing in the x direction is uniform then $x_{\xi\xi} = 0$ and furthermore since the grid is orthogonal then $y_{\xi} = 0$, which results in the vanishing of the additional terms above and thus the truncation error being $O(h^2)$. However, in general these additional terms do not vanish but can be shown to be proportional to the rate of change of the grid spacing and to the departure from orthogonality. Note that since the additional terms involve second derivatives of the solution variable, they constitute, in effect, a negative numerical diffusion. It is observed that unless care is exercised in generating the coordinate system, difference methods for first-order partial differential equations will be only first order and those for second-order equations will be inconsistent. However, the truncation error for first- and second-order derivatives can be increased to $O(h^2)$ and $O(h)$, respectively, if the second-order derivatives of x and y are $O(h^2)$. This, of course, limits the rate of change in grid spacing and the curvature of the coordinate lines.

52. Mastin further shows that the degree to which nonorthogonality

increases the truncation error is determined by the value of the Jacobian of the mapping. The factor that causes an increase in truncation error for first derivatives is derived to be $1/\sin(\theta - \phi)$ where ϕ and θ denote the angles of inclination of the η = constant and ξ = constant coordinate lines. This is exactly the factor by which the Jacobian is decreased by the shearing of the coordinate lines. A reasonable degree of nonorthogonality has a negligible effect; however, the rate of increase in truncation error increases as the grid becomes less orthogonal. It can be shown that when the angle between grid lines is greater than 45 deg the error from nonorthogonality is less than that from a nonuniform grid spacing. In addition, computations have demonstrated that doubling the grid spacing from one point to the next results in a 17 percent error in flow solutions. Thus the rate of change of the grid spacing and the departure from orthogonality must be controlled, or else significant error can be introduced.

53. Constraints. The above discussion on grid-induced errors has been given as a lead-in to the generation of adaptive grids using a variational approach. On the one hand, there is a need to force grid points to concentrate near large solution gradients; on the other hand, as illustrated in the previous discussion, there is a need to generate grids that are relatively smooth and don't deviate too much from being orthogonal. The calculus of variations is well-suited to handle such problems since integrals over the grid can be written which measure the three desired features discussed above, namely, (a) concentration of grid points near large solution gradients, (b) a smooth distribution of grid points, and (c) a relatively orthogonal grid. The variational formulation of adaptive grids is discussed in detail by Brackbill (1982). A brief discussion follows here.

54. The volume of computational cells in three dimensions (the area in two dimensions) is given by the Jacobian, J, of the mapping (Equation 6). Therefore, if the integral

$$I_w = \int_D w(x,y)\, J\, dV \,, \tag{39}$$

which is a measure of the weighted variation of the computational cell size over the grid, is minimized for some weight function $w(x,y)$, which is some measure of solution error, a concentration of grid points can be obtained. In

other words, where the weight function $w(x,y)$ becomes large the cell size becomes small, and where $w(x,y)$ becomes small the size of the computational cell becomes large.

55. Likewise, the smoothness of the grid is measured by the integral

$$I_s = \int_D \left[(\nabla\xi)^2 + (\nabla\eta)^2 \right] dV \tag{40}$$

with the orthogonality measured by

$$I_o = \int_D (\nabla\xi \cdot \nabla\eta)^2 \, J^3 \, dV \tag{41}$$

where the J^3 is added to cause orthogonality to be emphasized more in larger cells. Note that $\nabla\xi \cdot \nabla\eta = 0$ for an orthogonal grid. Therefore an adaptive grid generator can be developed by minimizing a sum of the three integrals given in Equations 39-41, i.e.,

$$I = \lambda_s I_s + \lambda_o I_o + \lambda_w I_w \tag{42}$$

56. From the above expressions for I_s, I_o, and I_w we have

$$I_s \sim \frac{N^2}{L^2} L^3 \; ; \quad I_o \sim \frac{L^5}{N^5} L \; ; \quad I_w \sim W \frac{L^3}{N^3} L^3$$

where

N = a characteristic number of points

L = a characteristic length

W = average weight function over the field

Thus, for Equation 42 to be dimensionally correct

$$\lambda_o = \lambda'_o \left(\frac{N}{L}\right)^7 \quad \text{and} \quad \lambda_w = \lambda'_w \left(\frac{N}{L}\right)^5 \frac{1}{W}$$

where λ'_o and λ'_w are positive constants. In two dimensions, the factors $(N/L)^7$ and $(N/L)^5$ both become $(N/L)^4$ since the Jacobian is then proportional to L^2/N^2 rather than to L^3/N^3. The characteristic length and number

of points might logically be taken as the cube roots of the volume and the total number of points in the field, respectively, in three dimensions, the square root being used in two dimensions.

57. Obviously, by selecting appropriate values of λ_s, λ'_w, and λ'_o, grids that emphasize smoothness, concentration of grid points, or orthogonality can be generated. The variational formulation thus provides a competitive enhancement of these three grid characteristics. A note of caution concerning orthogonality is perhaps needed. Purely orthogonal grids cannot be generated when prescribing the location of boundary points, even if the condition $\cdot = 0$ is enforced. Since derivatives of the coordinates have to be specified to satisfy orthogonality at the boundaries, specifying the location of the points overspecifies the problem with second-order partial differential equations as the grid generation system. Therefore, in order to generate a strictly orthogonal grid, the boundary points must be allowed to move on the boundary. Obviously, in many cases this constitutes a major restriction on the generation of a useful grid. Therefore large values of λ'_o compared with values of λ_s and λ'_w will only enhance the orthogonality of the grid. In general, the grid is still nonorthogonal and all terms in the governing transformed partial differential equations must be retained.

58. For a 2-D adaptive grid generator, partial differential equations for the solution of the physical coordinates (x,y) are desired. These are obtained through the calculus of variations applied to the minimization of the sum of integrals in Equation 42. In general, if we wish to find functions $\phi_i(x,y)$ that are differentiable on (x,y) and satisfy fixed constraints on the boundary of the domain that minimize some integral functional

$$I = \int_D f\left(x, y, \phi_i, \frac{\partial \phi_i}{\partial x}, \frac{\partial \phi_i}{\partial y}\right) dx\, dy \tag{43}$$

the calculus of variations gives that $\phi_i(x,y)$ are the solutions of the Euler-Lagrange equations

$$\nabla \cdot \left[\frac{\partial f}{\partial(\nabla \phi_i)}\right] - \frac{\partial f}{\partial \phi_i} = 0 \tag{44}$$

59. Two-dimensional generator. Brackbill (1982) has derived the 2-D

Euler-Lagrange equations for the minimization of Equation 42 as

$$b_1 x_{\xi\xi} + b_2 x_{\xi\eta} + b_3 x_{\eta\eta} + a_1 y_{\xi\xi} + a_2 y_{\xi\eta} + a_3 y_{\eta\eta} = -J^2 \frac{\lambda w}{2} \frac{\partial w}{\partial x}$$

$$a_1 x_{\xi\xi} + a_2 x_{\xi\eta} + a_3 x_{\eta\eta} + c_1 y_{\xi\xi} + c_2 y_{\xi\eta} + c_3 y_{\eta\eta} = -J^2 \frac{\lambda w}{2} \frac{\partial w}{\partial y} \quad . \tag{45}$$

For completeness, the coefficients are reproduced from the cited paper and presented below:

$$a_i = \lambda_s a_{s_i} + \lambda_w w a_{w_i} + \lambda_o a_{o_i} \quad ,$$

$$b_i = \lambda_s b_{s_i} + \lambda_w w b_{w_i} + \lambda_o b_{o_i} \quad , \text{ and}$$

$$c_i = \lambda_s c_{s_i} + \lambda_w w c_{w_i} + \lambda_o c_{o_i} \quad .$$

$$a_{s1} = -A\alpha \quad ; \quad b_{s1} = B\alpha \quad ; \quad c_{s1} = C\alpha$$

$$a_{s2} = 2A\beta \quad ; \quad b_{s2} = -2B\beta \quad ; \quad c_{s2} = -2C\beta$$

$$a_{s3} = -A\gamma \quad ; \quad b_{s3} = B\gamma \quad ; \quad c_{s3} = C\gamma$$

where

$$A = x_\xi x_\eta + y_\xi y_\eta \quad , \quad B = y_\xi^2 + y_\eta^2 \quad , \quad C = x_\xi^2 + x_\eta^2 \quad ,$$

and (46)

$$\alpha = \frac{x_\eta^2 + y_\eta^2}{J^3} \quad , \quad \beta = \frac{x_\xi x_\eta + y_\xi y_\eta}{J^3} \quad , \quad \gamma = \frac{x_\xi^2 + y_\xi^2}{J^3} \quad .$$

$$a_{w1} = -x_\eta y_\eta \quad , \quad b_{w1} = y_\eta^2 \quad , \quad c_{w1} = x_\eta^2 \quad ,$$

$$a_{w2} = x_\xi y_\eta + x_\eta y_\xi \quad , \quad b_{w2} = -2y_\xi y_\eta \quad , \quad c_{w2} = -2x_\xi x_\eta \quad ,$$

$$a_{w3} = -x_\xi y_\xi \quad , \quad b_{w3} = y_\xi^2 \quad , \quad c_{w3} = x_\xi^2 \quad ,$$

$$a_{o1} = x_\eta y_\eta \quad , \quad b_{o1} = x_\eta^2 \quad , \quad c_{o1} = y_\eta^2 \quad ,$$

$$a_{o2} = x_\xi y_\eta + x_\eta y_\xi \quad , \quad b_{o2} = 2(2x_\xi x_\eta + y_\xi y_\eta) \quad , \quad c_{o2} = 2(x_\xi x_\eta + 2y_\xi y_\eta) \quad .$$

$$a_{o3} = x_\xi y_\xi \qquad b_{o3} = x_\xi^2 \qquad c_{o3} = y_\xi^2$$

60. Equation 45 constitutes the adaptive grid generator in 2-D. Finite differences could be used to solve the equations on the transformed rectangular plane for values of (x,y) at each mesh point. These of course are then used to compute the metrics, e.g., x_ξ , y_η , etc., in the transformed differential equations (Equations 16-18). The solution of Equation 45 could be obtained at each time-step in an implicit fashion with the flow equations or, if desired, the grid generation could be lagged one or more time-steps behind the flow computations to yield an explicit grid generator. In addition, as previously discussed, interpolation of solution variables from the old grid to the new grid could be used to implement the adaptive grid or the implementation could be accomplished by transforming the time derivative in the governing equations.

61. An interesting development occurs if the coefficients in Equation 42 are set to be

$$\lambda_s = 1$$
$$\lambda_w' = \lambda_o' = 0 \tag{47}$$

to generate as smooth a coordinate system as possible. If the algebra is performed, it is seen that the grid generating system given by Equation 45 becomes

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0$$
$$\tag{48}$$
$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0$$

where

$$\alpha = x_\eta^2 + y_\eta^2$$
$$\beta = x_\xi x_\eta + y_\xi y_\eta \tag{49}$$
$$\gamma = x_\xi^2 + y_\xi^2$$

In other words, the generating system becomes the elliptic system given by Equations 5 and 6 where the P and Q control functions are set to zero, i.e., an elliptic system based upon Laplace's equation rather than Poisson's equation is used. Similarly, if $\lambda_w \neq 0$ the weighting function $w(x,y)$, which as noted is related to some measure of the solution error, can be related to the control functions in an elliptic system based on Poisson's equation.

62. One-dimensional generator. An adaptive grid generator based upon the variational approach can also be derived for 1-D problems, e.g., a surge moving down a river. If the coefficients in Equation 45 are evaluated for the case of x being the only dependent variable, the grid generation equation becomes

$$\left(\frac{\lambda_s}{J^3} + \lambda_w w \right) x_{\xi\xi} = - J^2 \frac{\lambda_w}{2} \frac{dw}{dx} \tag{50}$$

where the Jacobian, J, reduces to x_ξ. Neglecting the smoothness contribution, this equation can be reduced to

$$\frac{d(x_\xi)}{d\xi} = - x_\xi \frac{dx}{d\xi} \frac{1}{2w} \frac{dw}{dx} \tag{51}$$

or

$$\frac{d(x_\xi)}{x_\xi} = - \frac{1}{2} \frac{dw}{w} \tag{52}$$

Integrating Equation 52 then yields

$$\ln x_\xi = - \frac{1}{2} \ln w + \text{constant} \tag{53}$$

or

$$\ell n \left(x_\xi w^{1/2} \right) = \text{constant} \tag{54}$$

which implies

$$x_\xi w^{1/2} = \text{constant} \tag{55}$$

This is similar to the generating system below for the equidistribution of w discussed by Thompson (1983b)

$$x_\xi w(x) = \text{constant} \tag{56}$$

Therefore minimization of error by equidistribution is equivalent to solving a Euler-Lagrange equation arising from a variational approach.

63. Weight functions. As noted, the effect of the weight function, w, is to reduce the grid spacing where w is large, and therefore the weight function should be set as some measure of the solution error, or as some measure of the solution variation. The discussion to follow has been taken from Thompson, Warsi, and Mastin (1984). The simplest choice is just the solution gradient, e.g., in one dimension

$$w = u_x \tag{57}$$

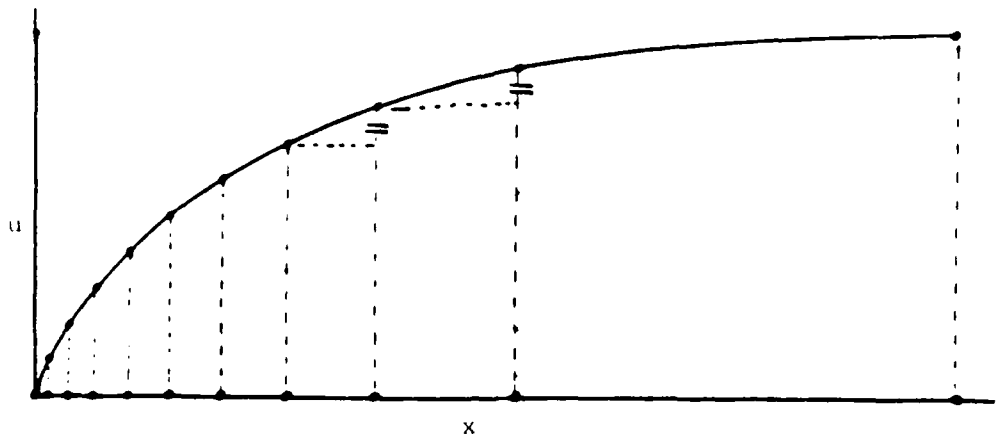
where u is some solution variable. In this case, Equation 56 becomes

$$x_\xi u_x = \text{constant}$$

which then reduces to

$$u_\xi = \text{constant}$$

With the solution gradient as the weight function, the point distribution thus adjusts so that the same change in the solution occurs over each grid interval, as illustrated below:



This choice for the weight function has the disadvantage of making the spacing infinitely large where the solution is flat, however.

64. A closely related choice, also based on the solution gradient, is the form

$$w = \sqrt{1 + u_x^2} \tag{58}$$

Now an increment of arc length, ds, on the solution curve, $u(x)$, is given by

$$ds^2 = dx^2 + du^2 = (1 + u_x^2)\, dx^2$$

so that this form of the weight function may be written

$$w = s_x$$

and then Equation 56 becomes

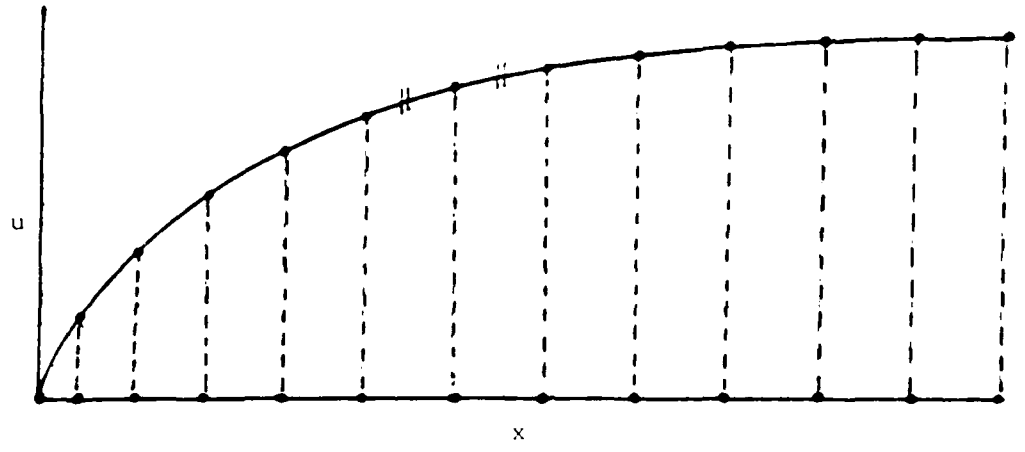
$$x_{\xi} s_x = \text{constant}$$

which reduces to

$$s_{\xi} = \text{constant}$$

Thus, with the weight function defined by Equation 58, the grid point distribution is such that the same increment in arc length on the solution curve

occurs over each grid interval. For the curve shown above this gives the following point distribution:



Unlike the previous choice, this weight function gives uniform spacing when the solution is flat. The concentration of points in the high-gradient region, however, is not as great.

65. This concentration can be increased, while still maintaining uniform spacing where the solution is flat, by altering the weight function to

$$w = \sqrt{1 + \epsilon^2 u_x^2} \tag{59}$$

where ϵ is a parameter to be specified. Now considering u to be plotted against x/ϵ, we have for an increment of arc length on this solution curve,

$$ds^2 = \left[d\left(\frac{x}{\epsilon}\right)\right]^2 + du^2 = \left[1 + u_{\left(\frac{x}{\epsilon}\right)}\right]\left[d\left(\frac{x}{\epsilon}\right)\right]^2$$

so that this weight function is equivalent to

$$w = s_{\left(\frac{x}{\epsilon}\right)}$$

and Equation 56 becomes

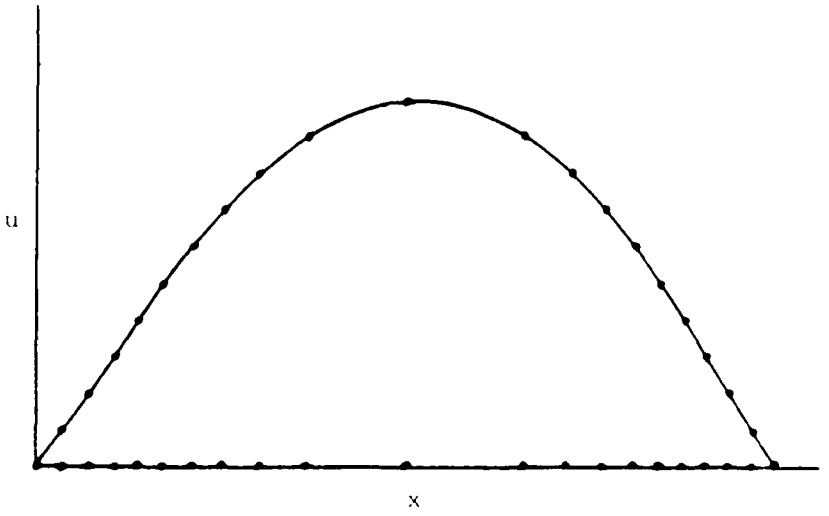
$$\left(\frac{x}{\epsilon}\right)_i - \bar{s}_{\left(\frac{x}{\epsilon}\right)} = \text{constant}$$

which reduces to

$$s_{,} = \text{constant}$$

Thus we have equal increments of arc length on the solution curve with u plotted against x/ϵ in this case. Now division of the abscissa by ϵ for a flat curve would simply reduce the spacing by the same factor. However, since the slope steepens as the curve is compressed to the left by this change of scale, the effect on the spacing where the curve is not flat will be a greater reduction in spacing. In fact, since the 1 in the weight function given by Equation 59 tends to produce equal spacing, while the $\alpha^2 u_x^2$ tends to produce concentration in the high-gradient regions, with infinite spacing in flat regions, this weight function involves a weighted average between the tendency toward equal spacing and that toward concentration entirely in the high-gradient regions. The larger the value of α, the stronger will be the concentration in the high-gradient regions and the wider the spacing in the flat regions.

66. A disadvantage of all the above forms of the weight function is that in regions near solution extrema, i.e., where $u_x = 0$ locally, spacing is similar to that in flat regions, as is illustrated below for the form given by Equation 57:



Although the distributions produced by the solution arc length forms, Equations 58 and 59, would have closer spacings near the extrema, the effect is still the same, i.e., to concentrate points only near gradients, not extrema.

67. Concentration near solution extrema can be achieved by incorporating some effect of the second derivative, u_{xx}, into the weight function. A logical approach is to include this effect through consideration of the curvature of the solution curve:

$$K = \frac{u_{xx}}{\left(1 + u_x^2\right)^{3/2}}$$

If the weight function is taken as

$$w = 1 + \alpha^2 |K| \tag{60}$$

points will be concentrated in regions of high curvature of the solution curve, e.g., near extrema, with a tendency toward equal spacing in regions of zero curvature, i.e., where the solution curve is straight (not necessarily flat). This weight function, however, has the serious disadvantage of treating high-gradient regions with little curvature essentially the same as regions where the curve is flat. Thus in the curve shown above, nearly all the points would be concentrated near the maximum in the curve, with very wide spacing in the high-gradient regions on both sides.

68. A combination of the weight functions given by Equations 59 and 60 provides the desired tendency toward concentration both in regions of high gradient and near extrema:

$$w = (1 + \beta^2 |K|) \sqrt{1 + \alpha^2 u_x^2} \tag{61}$$

where α and β are parameters to be specified. Clearly, concentration near high gradients is emphasized by large values of α, while concentration near extrema (or other regions of large curvature) is emphasized by large β.

69. Another approach to the inclusion of the second derivative is simply to take the weight function as

$$w = 1 + \alpha |u_x| + \beta |u_{xx}| \tag{62}$$

where α and β are nonnegative parameters to be specified. Note that the replacement of Equation 59 with the form given by Equation 62, with $\beta = 0$,

still leaves a reasonable form for the weight function, but the clear association with the geometric properties of the solution curve is lost. In this case, the weight function corresponding to Equation 58 would, after substitution in Equation 56, lead to the condition

$$x_\xi + u_\xi = \text{constant}$$

which corresponds to an equal distribution of the distance between points on the solution curve along a right-angle path formed by Δx and Δu from one point to the next. While this distance has some indirect relation to arc length on the solution curve (the chord length being the hypotenuse of the right triangle formed by this Δx and Δu), the direct association with arc length is preferable. Following the same reasoning, the use of solution curve curvature, rather than simply the second derivative, is also preferable. Therefore the form given by Equation 61 is probably more appropriate than that of Equation 62.

70. Since the numerical evaluation of higher derivatives can be subject to considerable computational noise, the use of formal truncation error expressions as the weight function is usually not practical, hence the emphasis above on solution gradients and curvature. Some problems may arise even with solution curvature, i.e., with second derivatives, in rough transits. It is common in any case to limit the grid point movement at each time-step and/or to smooth the new point distribution.

71. One final note on weighting functions might be offered. When more than one dependent solution variable is being computed, e.g., the water-surface elevation and u and v velocities in depth-averaged flow computations, it is not clear how to choose the most appropriate weight function. Some researchers have generated separate adaptive grids for solution variables, e.g., Dwyer, Smooke, and Kee (1982), whereas others such as Yanenko et al. (1982) have attempted to use weight functions that are composed of a dependence on several solution variables simultaneously. Additional comments on weight functions for free-surface hydrodynamic problems will be offered in PART IV.

Attraction-repulsion approach

72. In this approach to adaptive grid generation, the grid points move directly under the influence of mutual attraction or repulsion between points. The basic idea is to generate a grid speed for each grid point as a result of

the collective attraction, or repulsion, of all other points. This is accomplished by assigning to each point an attraction (repulsion) that is proportional to the difference between some measure of the local truncation error and a measure of the average truncation error over the complete grid. Thus points with a local error that exceeds the average error attract other points, with a consequent reduction in grid spacing and thus a reduction in local error, while points with a local error less than the average repel other points and thus increase the local grid spacing. In analogy with the electrostatic force between two electrical charges, the attraction exerted by a point on other points is attenuated by an inverse power of the distance between points in the computational region.

73. One-dimensional. For two points A and B, Anderson and Rai (1982) write the grid velocity at B due to error at point A in computational space as

$$V_{BA} = K \frac{|e|_A - |e|_{av}}{r_{BA}^n} \tag{63}$$

where

K = a proportionality factor chosen so that the velocity at any point does not exceed a preset maximum

$|e|$ = a measure of the truncation error

With $V_{BA} = \xi_t$, the grid speed in 1-D physical space becomes

$$x_t = \frac{V_{BA}}{\xi_x} \quad . \tag{64}$$

If the number of fixed points is N, a summation over all points yields the value of x_t to be inserted into the transformed flow equations with the time derivative transformed by Equation 19. A time integration of the grid speed equation then yields the physical location of the new grid points for use in computing the metrics of the transformation.

74. Two-dimensional. A discussion of a 2-D adaptive grid generator based upon the concept of point attraction or repulsion is also presented by Anderson and Rai (1982). As in the 1-D case, grid speeds x_t and y_t are determined by prescribing a certain grid function that is a function of $|e^{\xi}|$ and $|e^{\eta}|$ where [illegible] error estimates in the ξ

and η directions. A summation over all points then yields the final grid speed equations.

75. Error forms. As in the variational approach, some estimate or measure of the truncation error must be selected to cause a motion of the grid. The term $|e|$ in Equation 63 could, of course, be the actual truncation error; however, any other quantity could also be used. For example, $|e|$ could be taken to be any of the forms presented for the weighting function, w, in Equations 57-62. The same comments previously made concerning the use of higher derivatives of solution variables also apply. Therefore, if higher derivatives are used in the computation of $|e|$, large amounts of smoothing will probably be required.

76. Constraints. A disadvantage of using adaptive grid generators based upon the grid speed approach is that control of grid motion is not easily maintained. Such control is one of the major advantages of using the variational approach since control is inherently achieved through the specification of λ_s, λ_w, and λ_o in Equation 42. Although grid control is difficult, the point attraction approach will always prohibit the crossing of grid lines of the same family. This is illustrated by Equation 63. As two points approach each other, the local error becomes less than the average error and thus the force that drives the grid becomes negative, i.e., repulsive, resulting in the points moving apart. Attempts at controlling grid oscillations and non-smoothness of the grid have been made by damping the grid speeds permitted by limiting the change allowed in the Jacobian, i.e., cell size, at each point. This technique is discussed in detail by Anderson and Rai (1982).

Variable Number of Grid Points

77. Most of the research in adaptive grids has followed the approach of moving a fixed number of grid points in order to be able to accurately resolve steep fronts in flow computations, or perhaps concentration profiles, during transport processes. This approach utilizes coordinate transformations so that even though the grid points are moving in the physical plane all computations are performed in a fixed, rectangular, transformed plane. Another approach is to add or subtract grid points as the physics of the problem dictates. In either case, the grid movement in general is such as to equidistribute the truncation error or, as is normally the case, some measure of the truncation error, over the grid.

78. In the variable number of grid points approach, points are added or deleted until some specified maximum value of the error measure allowed over a grid interval is attained. Depending upon the value specified in the criterion to be satisfied, the variable node point approach may yield a more accurate solution than the fixed node point approach. However, it will normally result in more difficult coding and increased computational costs since the number of points will generally increase. Thus the most commonly held view of what constitutes adaptive grids is that a fixed number of points are made to move to better resolve field gradients. A combination of the two approaches is, of course, also possible.

Equidistributed quantities

79. As in the fixed number of grid points approach, some measure of the truncation error must be selected to be equidistributed over the grid. The grid generator of Dwyer, Smooke, and Kee (1982) is based upon equidistributing both the gradient of the dependent solution variables given by Z_i as well as the second derivatives, i.e., a mesh consisting of N points is sought such that

$$\int_{x_j}^{x_{j+1}} \left| \frac{dZ_i}{dx} \right| \leq \delta \left(\max_{0 \leq x \leq L} \left| Z_i \right| \right) \tag{65}$$

$$\int_{x_j}^{x_{j+1}} \left| \frac{d^2 Z_i}{dx^2} \right| \leq \gamma \left(\max_{0 \leq x \leq L} \left| \frac{dZ_i}{dx} \right| \right) \tag{66}$$

where $j = 1, 2, \ldots N - 1$ and $i = 1, 2, \ldots$ number of solution variables, δ and γ are arbitrarily selected small numbers, and $\max \rightarrow Z_i$ and $\max \rightarrow dZ_i/dx$ are determined from the solution on the previous grid. To ensure adequate smoothness of the grid, the grid spacing is bounded by

$$C \leq \frac{\Delta x_j}{\Delta x_{j-1}} \leq \frac{1}{C} \tag{67}$$

where C is some specified constant less than 1.0. Equations 65-66 in

essence define the weight functions that might be used in a variational approach employing a fixed number of points or the $|e|$ in the attraction-repulsion approach. Both steep gradient regions as well as regions of large curvature in any of the solution variables would be resolved by satisfying Equations 65-66.

Computational aspects

80. A new grid would be generated by proceeding through a number of steps. The governing partial differential equations involving the Z_i variables would first be solved on an extremely coarse uniformly spaced grid. The maximum values of $|Z_i|$ and $|dZ_i/dx|$ would then be computed and inequalities such as those in Equations 65-66 evaluated. If either is not satisfied in some interval, a grid point is inserted at the middle of the interval. After this mesh is generated, the inequality in Equation 67 is checked to ensure adequate smoothness. If Equation 67 is not satisfied in some interval, once again a grid point is inserted at the middle of the interval. The old solution is then interpolated onto the new grid and the steps described above are repeated. Dwyer, Smooke, and Kee (1982) indicate such a grid generator can be directly extended to two dimensions as long as sharp fronts align reasonably well with one of the coordinates. If not, techniques similar to those discussed by Anderson and Rai (1982) for aligning grids along shocks would have to be incorporated.

PART IV: APPLICABILITY OF ADAPTIVE GRIDS IN HYDRODYNAMIC MODELING

Basic Concerns

81. The ability of adaptive grids to reduce numerical error and thus enable the computation of more accurate solutions has been demonstrated. As previously noted, most of the research on the use of adaptive grids has been in the aerodynamics field where compressible flow solutions lead to the computation of large localized solution jumps, i.e., shock waves. Significant applications have also been made in flame-front problems. Before considering particular types of hydrodynamic problems for which numerical solutions are often sought, some basic concerns with the application of adaptive grids in hydrodynamic modeling are discussed.

Interpolation problems

82. Physical data. Most numerical hydrodynamic models are either 1- or 2-D depending upon the spatial averaging (Appendix A). Therefore the geometry of the water body is prescribed by a finite number of data points fixed in physical space. For example, in depth-averaged models a major data input is the water depth, usually prescribed at the center of finite difference computational cells. If the computational grid is allowed to move, obviously a new water depth must be prescribed at the center of each new grid cell. Therefore interpolation of the original water depth field is required each time the grid moves. Although such an interpolation should not result in additional solution error since interpolation from bottom contours is usually required to construct the original data set, it will increase the computational time. Another concern along these lines relates to bottom roughness coefficients, i.e., either Chezy or Manning coefficients. The normal procedure in hydrodynamic modeling of free-surface flows is to calibrate the model to known historical data through the variation of primarily the roughness coefficient. It would appear that such a calibration would have to take place on a fixed grid to generate the appropriate roughness coefficient field for interpolation to provide values on a moving or adaptive grid in other applications.

83. Solutions at fixed locations. In normal free-surface hydrodynamic modeling exercises, a major interest is in analyzing solution variables, e.g., the water-surface elevation, as a function of time at some fixed location such as a tide recording station. If adaptive grids are used, the locations of

points where the solution is generated continually move; thus once again interpolation at each time increment desired for plotting must be performed with a subsequent increase in the total cost of computations.

84. Separate grids. A major use of hydrodynamic models is to provide flow fields for use in water quality, sediment transport, and salinity intrusion studies. Depending upon the influence on the water density of the substance being transported and diffused, the flow and transport computations may or may not be coupled. In any case, an adaptive grid based upon the equidistribution of some error measure based upon the flow variables may not be a good grid to minimize numerical errors in the transport computations for some quality constituent. Dwyer, Smooke, and Kee (1982) experienced this problem in flame computations involving a burning spherical particle over which the flow had separated. In this calculation, both the flow field and the temperature gradients had to be well resolved. However, the coordinate system used in the temperature computations was not well-suited for the flow computations. As illustrated in Figure 4, Dwyer's approach consisted of using two different coordinate systems with interpolation between the two. In the past, hydrodynamic modelers have cautioned against the use of separate grids for flow and transport computations since the interpolation of the flow field, e.g., total depth and velocities in depth-averaged computations, results in mass not being strictly conserved. Mass conservation errors can result in large errors in transport-diffusion computations.

85. Another alternative is to use one grid but to have the adaption occur wherever needed by any variable. Thus there would be regions of point concentration due to the flow and other regions due to the transport. This would require the use of more grid points or else the multiple-point concentration regions would deplete the points available for the field at large. However, there would not be a need for interpolation as required for two separate grids.

86. Discontinuous grid movement. Unless the time derivatives in the governing differential equations of motion are transformed and grid computations are made at every time-step, interpolation of the numerical solution from the old grid to the new grid is required to initiate the computations every time a new grid is computed. Since the basic system of equations to be solved is hyperbolic in character, the effect of initial errors dies out fairly quickly in stable computations. Therefore interpolation of an adaptive grid may not be a major concern.

Computational costs

87. Grid generation. The typical time required on a CRAY computer to generate an initial coordinate system containing about 1,000 net points using Thompson's coordinate generation code (Thompson et al. 1974), is approximately 20 sec for 100-200 iterations of the Successive Over-Relaxation (SOR) solution scheme. This time, of course, varies for particular problems and for different degrees of coordinate control. It should be remembered that this generation code is based upon the variational approach where only smoothness is involved. Therefore the computational time required for a solution of the full generation equations given by Equation 45 would be substantially higher. However, it also should be remembered that the discussion above relates only to the initial grid generation, where more than likely the initial guess of the solution is a poor one. In subsequent computations for an adaptive grid, the initial guess would be the old grid. Normally, there will not be major changes between the old and new grids, so that perhaps only two to three iterations will be required for each subsequent grid generation. In other words, no more than 1 to 2 sec of computational time will be required for the generation of each new grid containing approximately 1,000 grid points.

88. It must be realized, however, that hydrodynamic calculations can often extend over long periods of time. For example, salinity intrusion computations on the Lower Mississippi River by Johnson and Boyd (in preparation) extended over 4 to 6 months with computational time-steps of 15 min, resulting in perhaps 15,000 computational steps requiring 30 to 45 min on a CRAY computer. If the grid ($\approx$3,000 points) had been computed at every time-step, an additional 10 to 20 hr of computer time would have been required. Obviously, this could not be tolerated and in fact would not have been needed. The movement of the salt front is such that if an adequate grid had been used the grid would probably not need to be computed more than once a day. This would then result in perhaps 15 min being required for the grid computations, i.e., perhaps one-third of the total computational costs would be attributed to the grid generation.

89. If an adaptive grid had been used in the study above, the governing equations, which were solved in a Cartesian form, would have to be transformed. This would probably result in an increase of perhaps two to three times the computational time cited, i.e., instead of 30 to 45 min the total computer time required for the flow and salinity computations would have been perhaps

1 to 2 hr. In this case, the cost of generating the adaptive grid would have been perhaps 10 to 15 percent of the total cost. For other more dynamic free-surface flow computations, e.g., storm surge computations, the percentage of the total cost would be much higher since the grid would need to be computed at virtually every time-step.

90. Time-step restrictions. If the time integration scheme for solving the governing partial differential equations is an explicit one, the maximum allowable time-step size will be controlled by the minimum size of the computational cells. Therefore, if an adaptive grid is used which results in an extremely small grid spacing in the vicinity of large gradients, a small time-step will be required which in turn results in more computer time for the flow computations than required for a fixed mesh with the same number of points. Of course to achieve essentially the same accuracy with a fixed grid, a fine grid spacing would have to be used over the entire grid. Thus the small time-step would still be required but now on a grid containing more points. Thus, if a certain accuracy is required, the use of an adaptive grid should result in less computing time. If implicit time integration schemes are utilized, there will not be stability restrictions on the time-step.

91. Higher order versus adaptive grids. Three approaches can be taken when attempting to compute more accurate numerical solutions of partial differential equations. Obviously, the number of grid points can be increased. However, if the number of grid points remains fixed either higher order solution schemes or adaptive grids with lower order schemes must be employed. A good example of a higher order scheme that filters out dispersive effects is the flux-corrected transport scheme of Zalesak (1979).

92. In its simplest terms, flux-corrected transport constructs the net transportive flux point by point as a weighted average of a flux computed by a low-order scheme and a flux computed by a high-order scheme. The crucial step is the determination of the weighting which is related to the flux limiting step. Basically, antidiffusive fluxes are computed as the difference between the high-order flux and the low-order flux and are then limited in such a way that the field is free of artificial extrema. An example of the superiority of the scheme over a low-order scheme is presented in Figure 5.

93. It is rather difficult to compare accuracy and computer time requirements of higher order schemes versus the use of adaptive grids with low-order schemes without coding both within the same computer code and applying

each to the same problem. However, experiences cited in the literature for each approach have been noted. Dortch* found that the use of Zalesak's flux-corrected transport scheme increased the computer time by a factor of 2.7 over the use of a donor cell scheme in a 2-D laterally averaged dye concentration computation. Approximately four times as many cells were required in the donor cell computation to achieve the same accuracy as those obtained with the flux corrected transport computations. Brackbill's 2-D computations of the steady advective diffusion equation showed that approximately 10 times as many cells in a fixed grid computation were required to achieve the same accuracy as that obtained with an adaptive grid. No comparison of computer time was given.

Hydrodynamic Applications

94. In the discussion below, several types of problems routinely addressed by numerical hydrodynamic models are noted. In some cases, adaptive grids would be of little use; whereas in others, a significant improvement in modeling capability could be realized through the use of adaptive grid techniques. Of the grid generation techniques developed to date, it is believed that the variational approach of Brackbill (1982) offers the most promise; therefore some appropriate weighting functions for such a grid generator are noted.

Flood routing

95. The computation of floods on natural rivers involves unsteady 1-D computations for the water surface and the flow rate, i.e. discharge. Although a wave is computed, it is a long-period wave (Figure 6) with a wavelength normally on the order of hundreds of miles. Therefore, in general, many grid points are contained within a wavelength and numerical error is small. For such problems, an adaptive grid would be of little use.

Dam breaks

96. Over the past five years, a major modeling activity within the Corps has been the computation of flooding resulting from hypothetical failures of Corps-regulated reservoirs. In such computations, there is basically a large single wave (Figure 7) moving down the valley below the dam. Major interest

---

* Personal communication, M. Dortch, 1984, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

is in predicting arrival times as well as maximum water-surface elevations at various locations. Theoretically, the use of a 1-D adaptive grid in such computations should result in increased accuracy for a fixed number of points. Of course the problem of interpolation of channel cross-sectional geometry data would add to the computational costs. In addition, if the cross section is broken into a conveyance plus a storage section, the storage volume on the overbank is the grid spacing times the storage cross-sectional area. As the dam-break peak moves downstream, the grid upstream will become sparse. As a result, it is easy to visualize continuity problems related to the storage volume occurring in the computations. Therefore, in a practical sense, it may be better to make computations on a grid containing enough points to yield the desired accuracy.

97. If an adaptive grid is employed in a dam-break model, a weight function based upon the water-surface elevation is suggested. Using Equation 61, the suggested weight function becomes

$$w = (1 + \beta^2 K) \sqrt{1 + \alpha^2 h_x^2} \tag{68}$$

with

$$K = \frac{h_{xx}}{\left(1 + h_x^2\right)^{3/2}}$$

where

h = water-surface elevation

α, β = positive parameters determined from numerical experimentation

Surge computations

98. One-dimensional. The sudden release of large volumes of water either from hydropower peaking operations or perhaps as a result of lock operations can create monoclinical gravity waves that are normally referred to as surges. Positive surges that occur downstream of such operations elevate the water surface, whereas negative surges that occur upstream depress the water level. The study of surges is important because of their possible adverse influence upon navigation, as well as the sudden large loads they can impose upon lock miter gates and appurtenances. Surges of large amplitude cannot be tolerated in practical navigation channels; therefore accurate predictive computations are essential.

99. The computation of such surges involves solving the 1-D unsteady flow equations presented in Appendix A. Many such models have been developed (Johnson, in preparation). However, a deficiency of all such models is the numerical dispersion and dissipation that is inherent when attempting to compute waves that are resolved over only a few grid points, e.g., <10 . Such problems are ideally suited for adaptive gridding, especially if a single surge is created. However, hydropeaking operations often create more than one surge. Figure 8 is an example of typical surges created downstream of Barkley Dam. Although multiple surges will not introduce particular problems with applying adaptive grids, more total points will be required so that there will be enough points to get concentration in each surge without depleting the field elsewhere. As a surge passes out the downstream boundary, of course more points become available elsewhere. The more surges expected in the field at any one time, the greater will be the computational time since the total number of points will be dependent upon the number of surges.

100. As with the dam-break problem, interpolation of the cross-sectional geometry tables would be required. However, the problems connected with overbank storage would probably not be as prevalent since overbank areas would not be as likely to be flooded. Since water velocity and surge height are major considerations in assessing the impact of surges on navigation, adaption could probably be based upon either. Once again the weight function given by Equation 68 is suggested if adaption is based upon the surge height; whereas the water-surface elevation would be replaced in Equation 68 by the average cross-sectional flow velocity if adaption to the velocity is desired.

101. Two-dimensional. A common numerical modeling exercise in coastal areas is the computation of storm surges using simplified forms of the depth-averaged equations presented in Appendix A. Such computations are required in storm insurance studies. As with the [illegible] discussed above, a common problem concerns being able to place enough points in a wavelength to decrease the numerical dissipation and dispersion [illegible]. Figure 9 is a plot of surge heights at three coastal sites [illegible] from Hurricane Camille. Once again adaptive gridding should [illegible] accurate solution. A grid generator based upon Equation 56 is to be employed with the weight function taken as

$$w = [illegible] \sqrt{[illegible]} \tag{69}$$

where

$$K = \frac{\nabla^2 h}{\left[1 + (\nabla h)^2\right]^{3/2}}$$

Short waves in harbors

102. The behavior of short waves in shallow water is a major concern of coastal engineers. The design of harbors and other facilities necessitates a detailed knowledge of short-wave characteristics since these waves attack harbor structures while being refracted, diffracted, and reflected through the harbor. An interesting computation by Brackbill (1982) involving aerodynamic wave reflections on an adaptive grid is presented in Figure 10. Such waves also influence navigation in the harbor area directly and through the currents they can induce.

103. In order to compute the behavior of such waves, additional terms must be added to the depth-averaged equations presented in Appendix A to account for the effect of vertical accelerations. The introduction of these terms presents problems in that solutions of the equations are quite sensitive to the numerical errors introduced by standard numerical techniques unless perhaps 10 to 20 grid points are used per wavelength. However, from a practical standpoint, there is a need to describe any short wave with only a few grid points. As with the surge computations, it would appear that adaptive grids would be the solution to the problem. However, as illustrated in Figure 11, many short waves are generally contained within the computational grid. Thus it is doubtful that an adaptive grid would yield much benefit since the total number of points required for good concentration over each wave would probably be about the same as that required with a fixed grid. A better approach might be to employ some higher order scheme, such as the flux-corrected transport previously discussed, applied on a grid containing as many points as economically possible.

Tidal circulations

104. Vertically averaged numerical models are often employed to compute flow fields in relatively shallow, well-mixed estuaries and coastal areas for use in water quality and sedimentation studies. Examples are the models by Butler (1980) and Norton and King (1977). Although the propagation of a tidal wave is computed, such waves are long-period waves that generally pose no

particular problems since several grid points lie within a wavelength. Although large gradients in the flow do occur when moving from deep areas to relatively shallow ones, e.g., from navigation channels to the shallow overbanks (Figure 12), it is not believed that in general a moving grid would be of use since basically the regions of high gradients are known and don't arbitrarily develop in the computational grid. However, it is believed that adaptive grid techniques would be of great use in numerically generating the initial grid.

105. Navigation channels. In Thompson's grid generation code, control functions can be prescribed to force a small grid spacing in particular regions. However, the control is relative to either ξ or η lines in the computational field. Unless the line to which attraction is desired lies on a boundary, its position in physical space is not known until after the grid has been generated. Attraction to specific lines in space is more difficult and more expensive and has not proved to be as successful. In other words, currently it is not possible with this code to force a fine grid spacing along winding navigation channels fixed in physical space. The generation of an adaptive grid which would only be generated once should provide a solution to this problem. Equation 45 would constitute the grid generator with the weight function, w, being prescribed so as to force a small grid spacing where w is large and a large spacing where w is small. Since navigation channels are defined by their increased depth, the appropriate weight function should be the water depth itself, i.e.,

$$w(x,y) = h(x,y) \tag{70}$$

where $h(x,y)$ is the total water depth relative to some mean water level. This would have the effect of generating a grid with the same volume in each vertical column.

106. Flooding and drying. As computations are made over a tidal cycle, quite often there is a need to allow for flooding of tidal flats on the flood cycle of the tide, with a subsequent drying of those areas on the ebb portion. In other words, the horizontal water boundary moves over a tidal cycle. One could think in terms of using adaptive gridding to handle such moving boundaries. An example of a moving boundary adaptive grid is presented in Figure 13. However, if the areas flooded and dried are essentially internal

areas, e.g., an island that appears and then disappears, some technique for allowing points to collapse into a single point when the area is flooded would be required. In addition, moving points around to better represent the water boundary would probably be to the detriment of the channel resolution. As a result of these anticipated problems, it is believed that the best approach would be to use adaptive grid techniques to generate the initial grid, with grid lines generally following bottom elevation contours. Wetting and drying could then be carried out on this grid as is currently done on Cartesian grids, i.e., by removing or adding cells from the computations by checking water-surface levels.

107. Finite element stability. On occasion, numerical experience confirms that the Norton-King (1977) finite element model used within the Hydraulics Laboratory can become unstable midway in a model execution due to the particular grid used. The current "cure" is to redistribute nodes or channel topography, in a manner based on experience, and execute the code again. For a mathematically consistent algorithm, the approximation error is always bounded by some measure (size) of the mesh (grid), i.e., the error is controlled by refining (perhaps locally) the grid. An improvement to the utilization of the codes, regarding tidal cycle restarts, could result from extension of the theory to include node motion. An energy norm distribution, say in h, the free-surface height, could detect regions of incipient non-smoothness (growing instability) and hence pinpoint regions in which a refined grid could be appropriate. Within the code itself, this amounts to adding the grid convection velocity terms, $\{\dot{X}\}$ in Equation 33.

Stratified flow

108. The computation of flow fields in bodies of water where the density varies over the depth often involves internal density currents. Such stratified flow refers to motions involving fluid masses of the same phase. A heavier liquid flowing beneath a lighter liquid, or a heavier gas moving under a lighter gas, will be subject to gravitational effects that depend upon the differences between the two densities. Figure 14 illustrates such a flow field for a cold-water inflow near the bottom of a flume. The internal motions in reservoirs due to temperature variations and salinity intrusion into relatively deep estuaries are examples of commonly computed stratified flow fields.

109. The salinity intrusion study by Johnson and Boyd (in preparation) on the Lower Mississippi River is a good example of numerical computations

involving density effects. The governing equations solved are the laterally averaged equations for both the flow field and the salinity field presented in Appendix A. Since the salinity computations involve the computation of a moving salt wedge with a sharp interface separating the salt water from the fresh water flowing down the river above the wedge, adaptive gridding should enable the computation of a much sharper front.

110. As in the Dwyer, Smooke, and Kee (1982) flame problem, it may be that separate grids would have to be employed for the flow and salt computations. However, due to the interpolation problems previously discussed, before more than one grid is employed, the use of a single grid should first be attempted. Since gradients in both the horizontal velocity and the salt concentration are large in the vicinity of the saltwater-freshwater interface, an appropriate weighting function to generate a single grid should involve both. For example, the following weight function might be applied:

$$W(x,z) = \alpha\sqrt{1 + \nabla U^2} + (1 - \alpha)\sqrt{1 + \nabla s^2} \tag{71}$$

where

α = constant with values between 0 and 1

U = horizontal velocity

s = salinity

Hydraulic jumps

111. When bottom slopes change from being steep enough to support supercritical flow to milder slopes, subcritical flow conditions are arrived at through the formation of a hydraulic jump. This is completely analogous to the formation of shock waves in compressible aerodynamic flow fields. Such flow conditions are rare in nature but can occur downstream of spillways at flow-control structures. The analysis of this free-surface hydrodynamic problem is normally accomplished with small-scale physical models. However, since the formation of shock waves is now routinely computed in aerodynamics, the numerical computation of the formation of hydraulic jumps should also be possible. In fact, Figure 15 presents the results of a preliminary computation of a hydraulic jump by Baker* using a finite element code. Obviously,

* Personal communication, A. J. Baker, 1983, University of Tennessee, Knoxville, Tenn.

use of an adaptive grid in such computations would be of great benefit; and the experience of compressible aerodynamic modelers would be directly applicable. In a similar fashion to the pressure gradient being used in the weighting function in many aerodynamic adaptive grid applications, the appropriate weighting function should probably be based on the gradient of the water depth.

PART V: SUMMARY

112. Since numerical solutions of the partial differential equations governing the hydrodynamics of bodies of water are obtained on grids that represent a discretization of the continuous physical domain, errors that are a function of the discretization invariably arise. Historically, such errors have been controlled in one of two ways. Either a complex higher order solution scheme is employed, or an extremely fine resolution grid is constructed over the physical domain so that simple low-order schemes can be used. From a programming and computational efficiency viewpoint, low-order schemes are more desirable. However, increasing the number of computational points is often an inefficient approach since portions of the domain may contain small gradients of the solution variables and a fine grid is not required there. The solution to this problem is to employ a grid that is free to move as the solution is computed such that a fine grid spacing occurs in regions of large solution gradients with a corresponding larger grid spacing in small gradient regions. Such grids are referred to as adaptive grids.

Conclusions

113. Adaptive grids can be implemented in both finite element solution schemes and finite difference solution schemes applied on boundary-fitted coordinates. The most natural implementation involves transforming the time derivative in the governing flow equations so that computations are performed in a fixed transformed grid with the influence of the grid movement in the physical plane entering the computations through terms involving the speed of the grid points.

114. The usual definition of an adaptive grid is one containing a fixed number of points, with the movement of those points dictated by the physics of the problem being solved. Several methods for generating such grids exist and there are advantages and disadvantages to each. However, all basically are concerned with the equidistribution of some measure of the solution error over the grid. The approach that seems to be the most logical, and that offers the greatest inherent control over the grid, is a variational approach developed by Brackbill (1982). A functional whose value is given by an integral equation

involving integrals related to grid smoothness and orthogonality, as well as the concentration of grid points, is minimized to yield differential equations for computing the grid point locations. Therefore different emphasis can be placed upon smoothness, orthogonality, and/or point concentration merely by changing the value of three constants. A major question is what to use for the weight function that controls the grid adaption.

115. Several hydrodynamic problems that are often addressed with numerical models have been discussed with respect to the usefulness of obtaining solutions on adaptive grids, and suggestions for weight functions have been given. In general, if the flow solution contains perturbations that either develop and/or move in arbitrary regions of the physical domain, adaptive gridding will be of greatest benefit, e.g., the computation of surges as a result of either locking or hydropower peaking operations, and the computation of hydraulic jumps and concentration fronts. However, if many perturbations exist in the domain at any particular time, adaptive gridding probably offers no advantage. The propagation of short waves in coastal areas is an example.

Recommendations

116. Much of the numerical modeling of free-surface hydrodynamics within the Corps of Engineers involves the depth-averaged computation of circulation patterns in estuaries and coastal areas. In such problems, a major concern is the representation of winding navigation channels. Although time-dependent adaptive grids don't in general appear to be required in the solution of such problems, a grid generator based upon the Brackbill variational approach should be an ideal generator for the initial grid. Resolution in the navigation channel could be achieved by adapting to the water depth with a resulting larger spacing in the nonchannel areas. Such a grid-generating model is required before the finite difference models of Johnson (1980) and Sheng and Hirsh (1984), based upon boundary-fitted coordinates, can be generally applied. In addition, it should also make the construction of finite element grids a much more automated process. Therefore it is recommended that a research effort be initiated to develop an estuarine/coastal fixed-grid generator model based upon adaptive grid techniques.

117. Although a general adaptive grid is not needed in tidal circulation models, numerical experience confirms that the depth-averaged finite

element model (RMA-2) often employed by the Corps can become unstable midway in a model execution if a grid that does not adequately resolve flow gradients in localized regions is employed. Research into how to best incorporate adaptive grid techniques into the existing model to automatically make minor local changes in the finite element grid will provide the potential for tremendous savings in computer time and, therefore, is highly recommended.

118. At the present time, small-scale physical models are employed to address the design of spillways where hydraulic jumps can occur. The numerical techniques employed in the development of models for computing the transition flow from supersonic to subsonic (and shock waves that occur) should be utilized to develop hydrodynamic transition flow models. Adaptive grids will enable such computations to be made more accurately at less cost.

REFERENCES

Anderson, D. A., and Rai, M. M. 1982. "The Use of Solution Adaptive Grids in Solving Partial Differential Equations," Numerical Grid Generation, ed. Joe F. Thompson, North-Holland.

Baker, A. J. 1983. Finite Element Computational Fluid Mechanics, McGraw-Hill/Hemisphere, New York.

Blumberg, Alan F., and Herring, H. James. 1983. "Circulation Modeling Using Curvilinear Coordinates," Dynalysis of Princeton, Princeton, N. J.

Brackbill, J. U. 1982. "Coordinate System Control: Adaptive Meshes," Numerical Grid Generation, ed. Joe F. Thompson, North-Holland.

Butler, H. L. 1978 (Dec). "Numerical Simulation of the Coos Bay-South Slough Complex," TR H-78-22, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

_____. 1980. "Evolution of a Numerical Model for Simulation of Long-Period Wave Behavior in Ocean-Estuarine Systems," Estuarine and Wetland Processes, eds. P. Hamilton and K. Macdonald, Plenum Press, New York.

Dwyer, H. A., Smooke, M. D., and Kee, R. J. 1982. "Adaptive Gridding for Finite Difference Solutions to Heat and Mass Transfer Problems," Numerical Grid Generation, ed. Joe F. Thompson, North-Holland.

Fread, D. L. 1978. "The NWS Dam-break Flood Forecasting Model," National Weather Service, Silver Spring, Md.

Johnson, B. H. 1978. "Mathematical Hydrodynamic Model of the Cumberland River to Aid in Physical Model Navigation Tests," In-house Report, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

_____. 1980 (Sep). "VAHM - A Vertically Averaged Hydrodynamic Model Using Boundary-Fitted Coordinates," Miscellaneous Paper HL-80-3, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

_____. 1981 (Feb). "A Review of Numerical Reservoir Hydrodynamic Modeling," Technical Report E-81-2, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

_____. 1982 (Aug). "Development of a Numerical Modeling Capability for the Computation of Unsteady Flow on the Ohio River and Its Major Tributaries," Technical Report HL-82-20, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

_____. "User's Guide for Branched Implicit River Model (BIRM) with Application to the Lower Mississippi River," (in preparation), US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Johnson, B. H., and Boyd, M. B. "A Numerical Study of the Impact on Salinity Intrusion of Deepening the Lower Mississippi River Navigation Channel" (in preparation), US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Johnson, B. H., Thompson, J. F., and Stein, A. B. 1982. "Modeling of Flow and Conservative Substance Transport on a Boundary-Fitted Coordinate System," Proceedings of Refined Modeling of Flows," IAHR, Paris, France.

Leendertse, J. J. 1967. "Aspects of a Computational Model for Long-Period Water Wave Propagation," RM-5294-PR, The Rand Corporation, Santa Monica, Calif.

Mastin, C. W. 1982. "Error Induced by Coordinate Systems," Numerical Grid Generation, ed. Joe F. Thompson, North-Holland.

Norton, W. R., and King, I. P. 1977. "Operating Instructions for Computer Program RMA-2 - A Two-Dimensional Finite Element Program for Problems in Horizontal Free Surface Hydromechanics," prepared for Association of Bay Area Governments Environmental Management Plan Bay Modeling Project, Resource Management Associates, Lafayette, Calif.

Reddy, R. N., and J. F. Thompson. 1977. "Numerical Solution of Incompressible Navier-Stokes Equations in the Integro-Differential Formulation Using Boundary-Fitted Coordinated Systems," AIAA 77-650, AIAA 3rd Computational Dynamics Conference, Albuquerque, N. M.

Sheng, Y. P., and Hirsh, J. E. 1984. "Numerical Solution of Shallow Water Equations in Boundary Fitted Grid," prepared for CERC, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Thompson, J. F. 1983a. "A Boundary-Fitted Coordinate Code for General Two-Dimensional Regions with Obstacles and Boundary Intrusions," Technical Report E-83-8, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Thompson, J. F. 1983b. "The Construction and Use of Adaptive Grids," Short Course on Finite Element Analysis in Fluid Mechanics and Heat Transfer, University of Tennessee, Knoxville, Tenn.

Thompson, J. F. 1983c. "A Survey of Grid Generation Techniques in Computational Fluid Dynamics," to be published in AIAA Journal, 1984, AIAA 21st Aerospace Science Meeting, Reno, Nev.

Thompson, J. F. 1984. "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations," AIAA 17th Fluid and Plasma Dynamics Conference, Snowmass, Col.

Thompson, J. F., and Mastin, C. W. 1983. "Order of Difference Expressions on Curvilinear Coordinate Systems," Advances in Grid Generation, ASME Fluids Engineering Conference, Houston, Tex.

Thompson, J. F., et al. 1974. "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, Vol 15, No. 3.

Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. 1983. "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review," Journal of Computational Physics, Vol 47, pp 1-108.

_____. 1984. Numerical Grid Generation, to be published by North-Holland.

Waldrop, W. R., and Farmer, R. C. 1976. "A Computer Simulation of Density Currents in a Flowing Stream," International Symposium on Unsteady Flow in Open Channels.

Waldrop, W. R., and Tatom, F. B. 1976. "Analysis of the Thermal Effluent from the Gallatin Steam Plant During Low River Flows," Report No. 33-30, Tennessee Valley Authority.

Wanstrath, J. J. 1977 (Sep). "Nearshore Numerical Storm Surge and Tidal Simulation," TR H-77-17, US Army Engineer Waterways Experiment Station, Vicksburg, Miss.

Ward, G. H., and Espey, W. H. 1971. "Estuarine Modeling: An Assessment," TRACOR, Inc., Austin, Tex., for the Water Quality Office, Environmental Protection Agency.

White, A. B., Jr. 1982. "On the Numerical Solution of Initial Boundary-Value Problems in One Space Dimension," SIAM J. of Numerical Analysis, p 683.

Yanenko, N. N., et al. 1979. Comput. Method. Appl. Mech. Eng., Vol 17/18.

Zalesak, Steven, T. 1979. "Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids," Journal of Computational Physics, Vol 31.

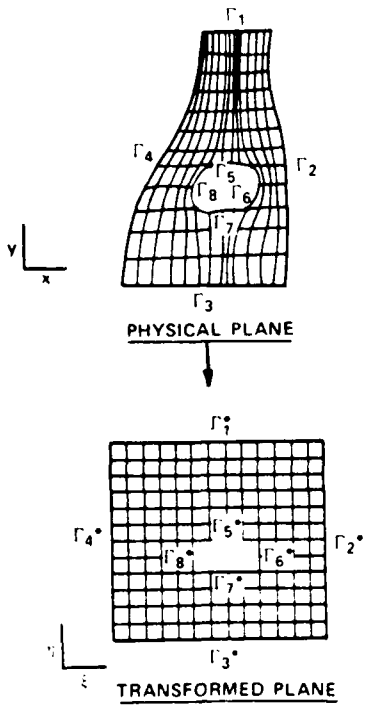


Figure 1. Transformation of physical domain to computational (note treatment of island)

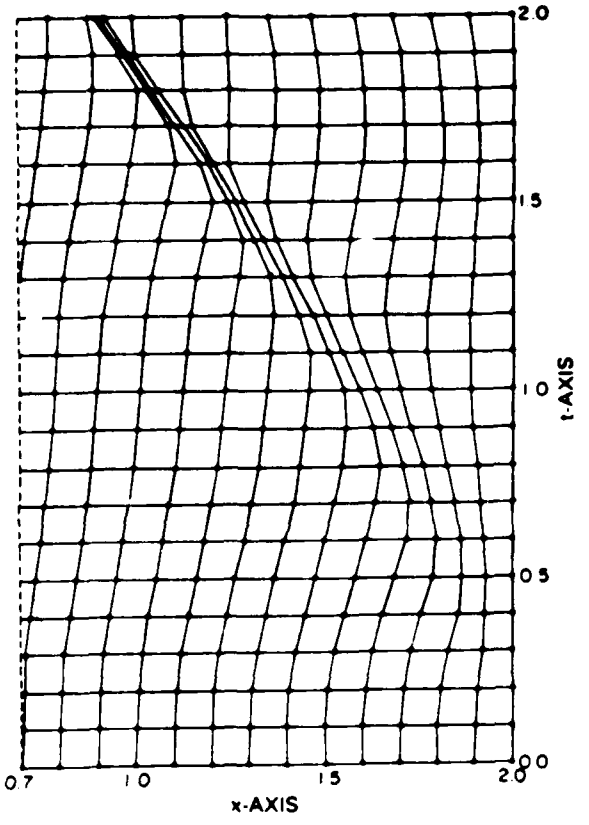
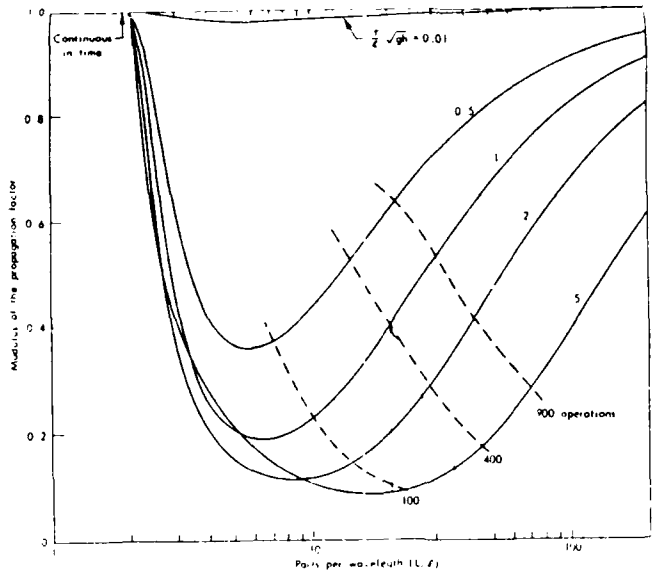
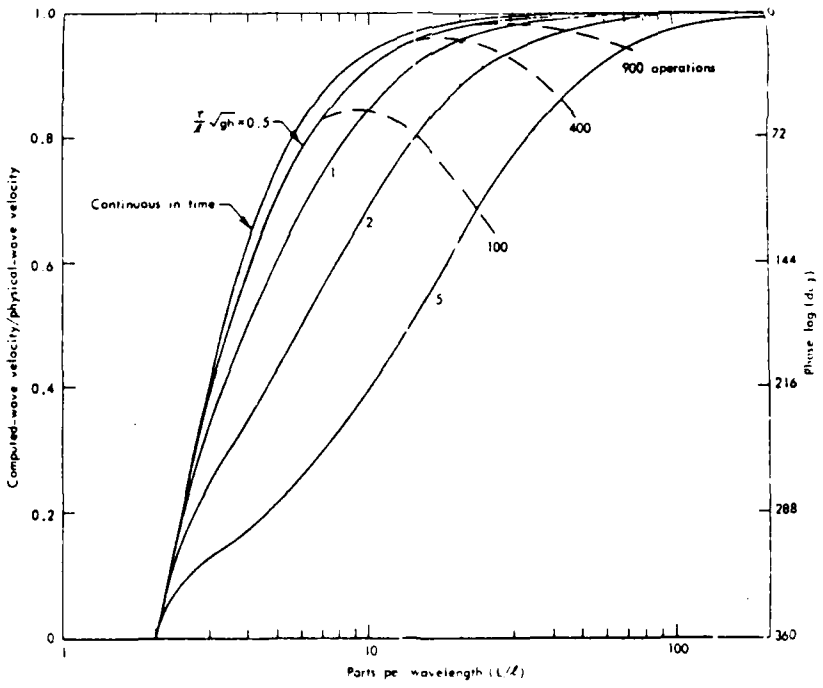


Figure 2. Example of an 1-D adaptive grid (from White 1982)

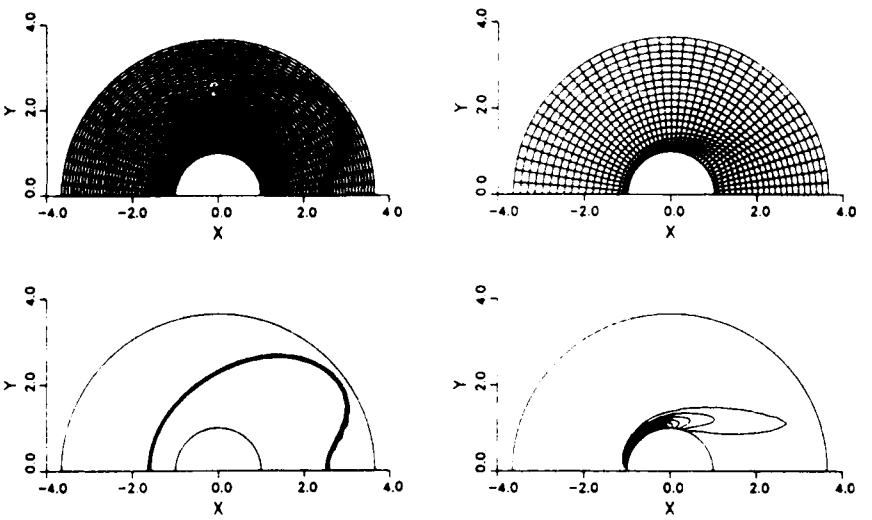


a. Dissipative error



b. Dispersive error

Figure 3. Effect of grid spacing and time-step on numerical solution (from Leendertse 1967)



a. Coordinate system for temperature computations and isotherm distribution

b. Coordinate system for flow computation and vorticity distribution

Figure 4. Example of multiple coordinate systems (from Dwyer, Smooke, and Kee 1982)

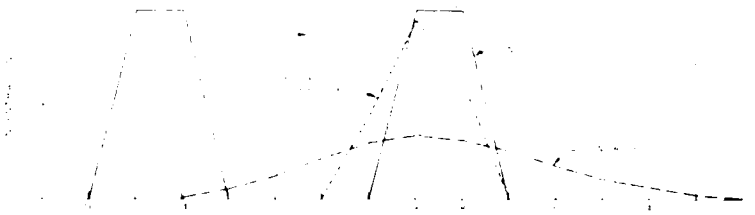


Figure 5. Idealized comparison of flux-corrected transport and donor cell (from Johnson, Thompson, and Stein 1982)

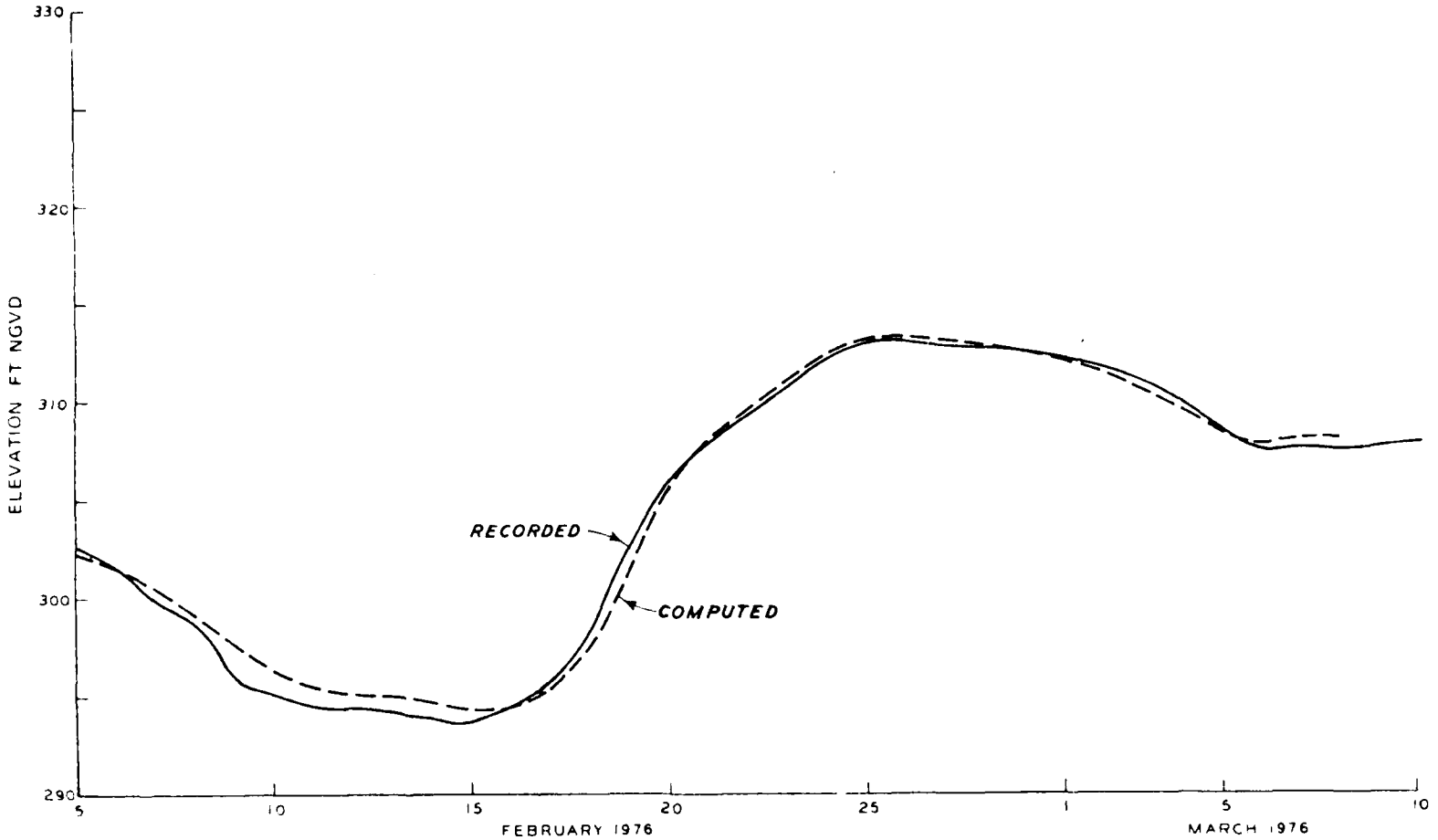


Figure 6. Example of long-period flood wave (from Johnson 1982)

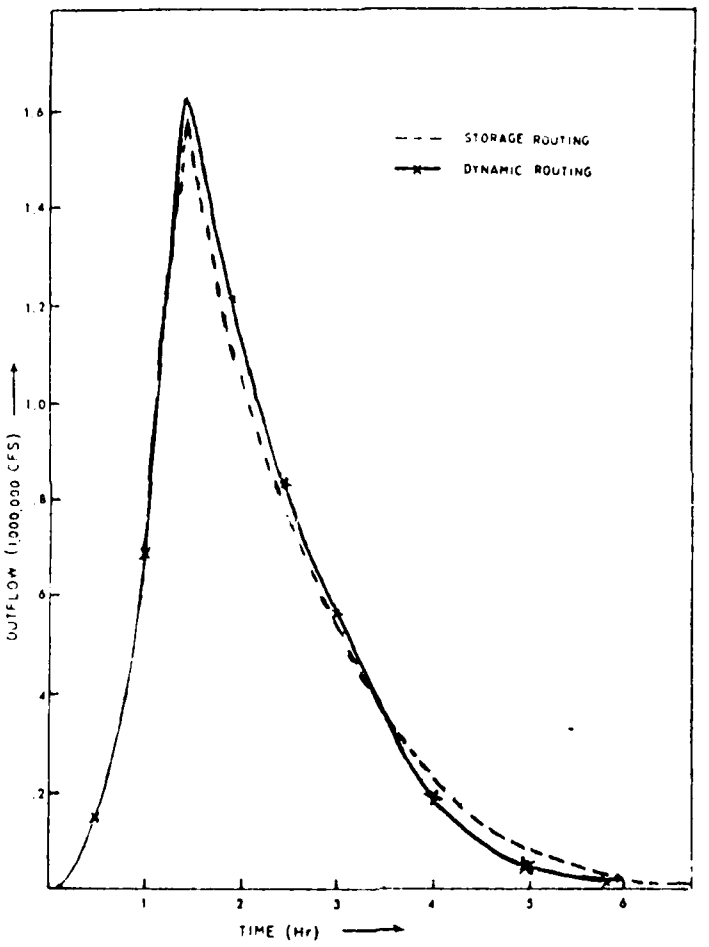
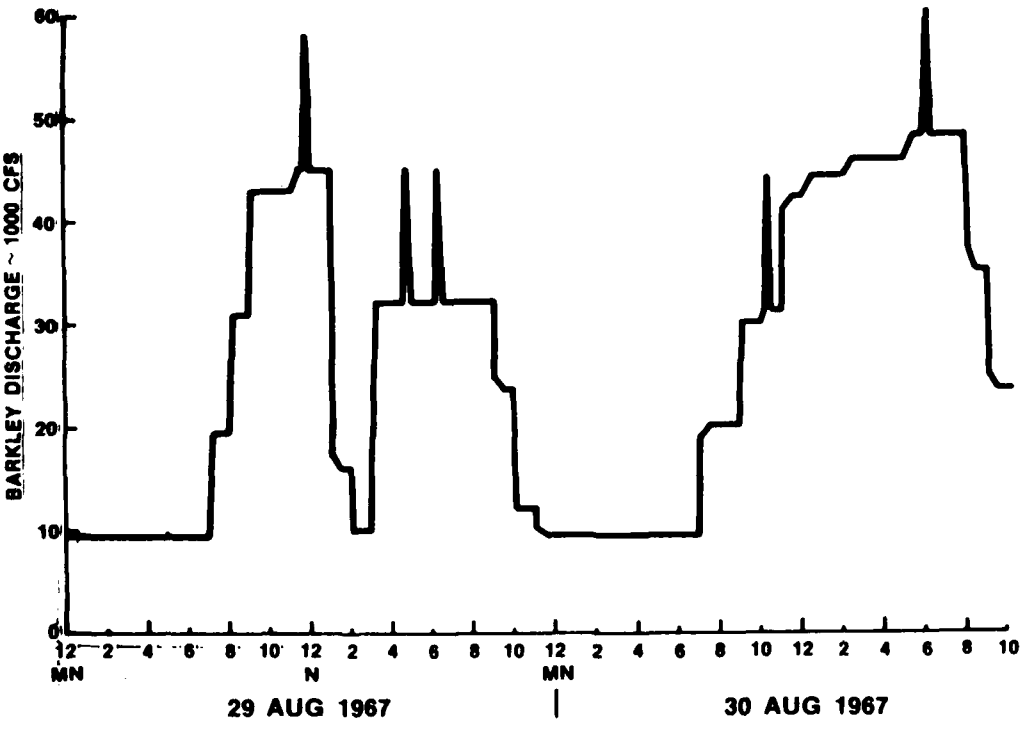
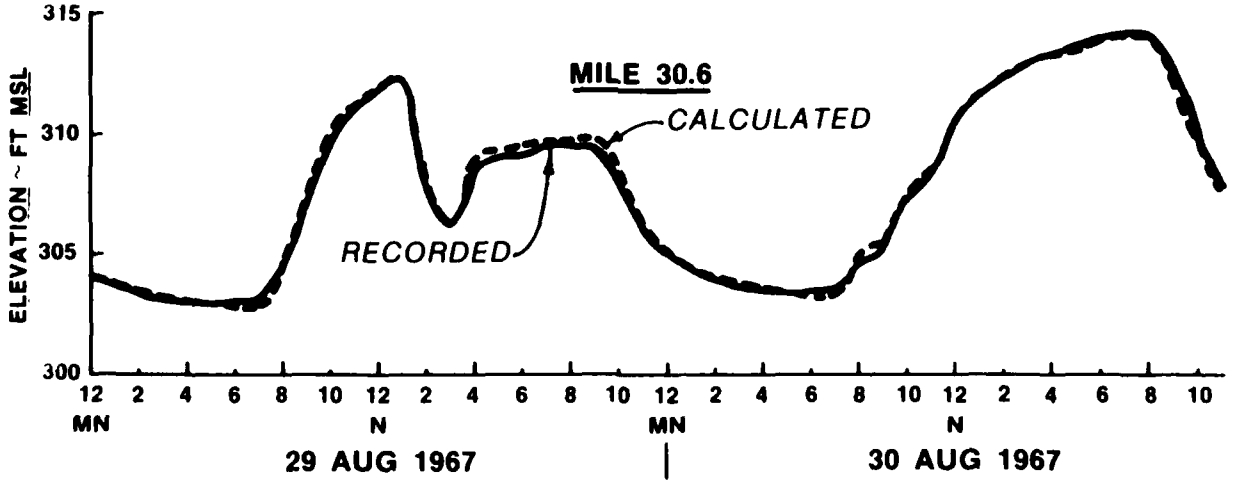


Figure 7. Outflow hydrograph from Teton Dam failure (from Fread 1978)



a. Discharge hydrograph



b. Downstream surge

Figure 8. Surge resulting from hydropower peaking at Barkley Dam (from Johnson 1978)

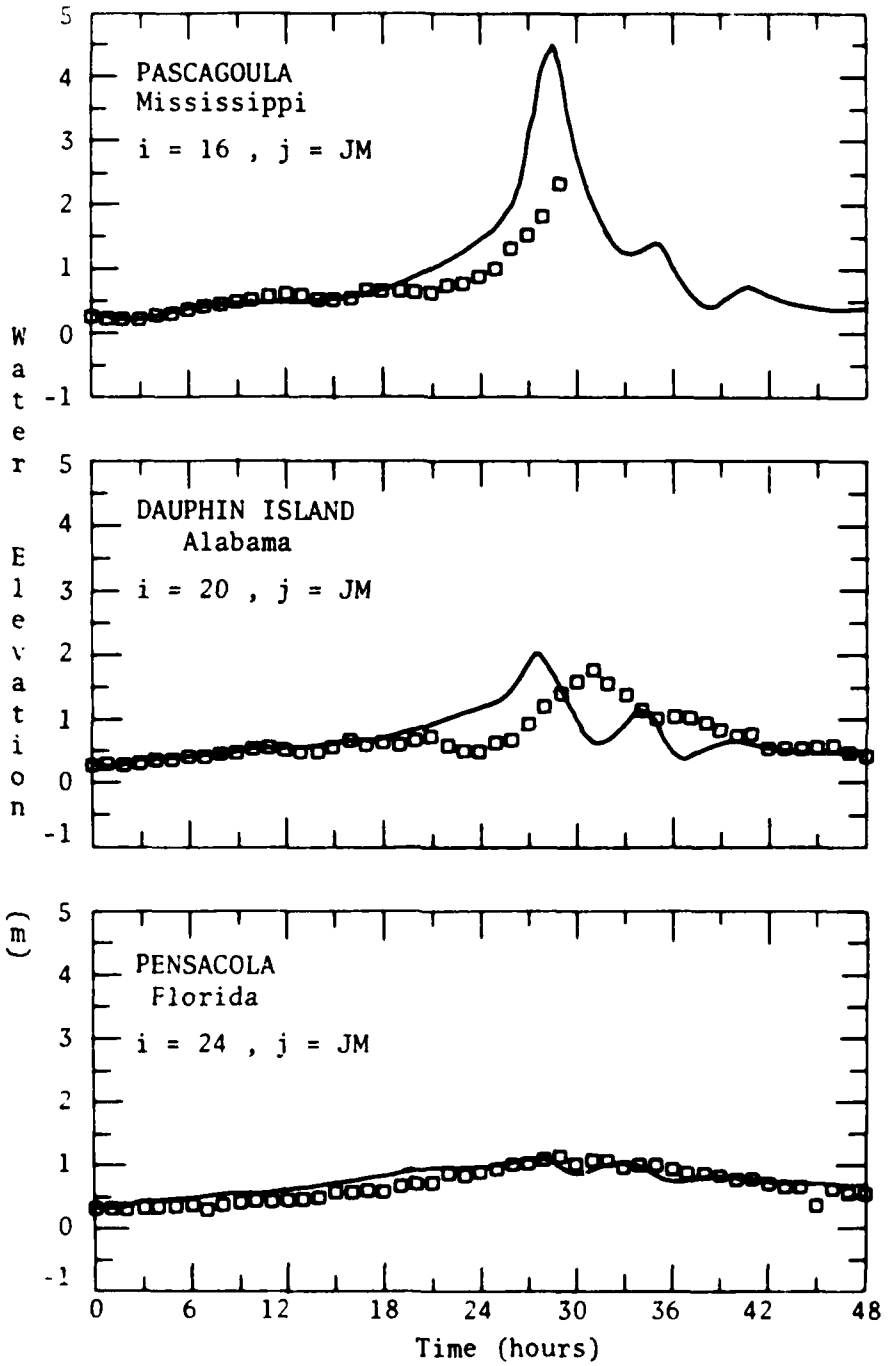


Figure 9. Observed (squares) and computed (solid) water levels for Hurricane Camille (from Wanstrath 1977)

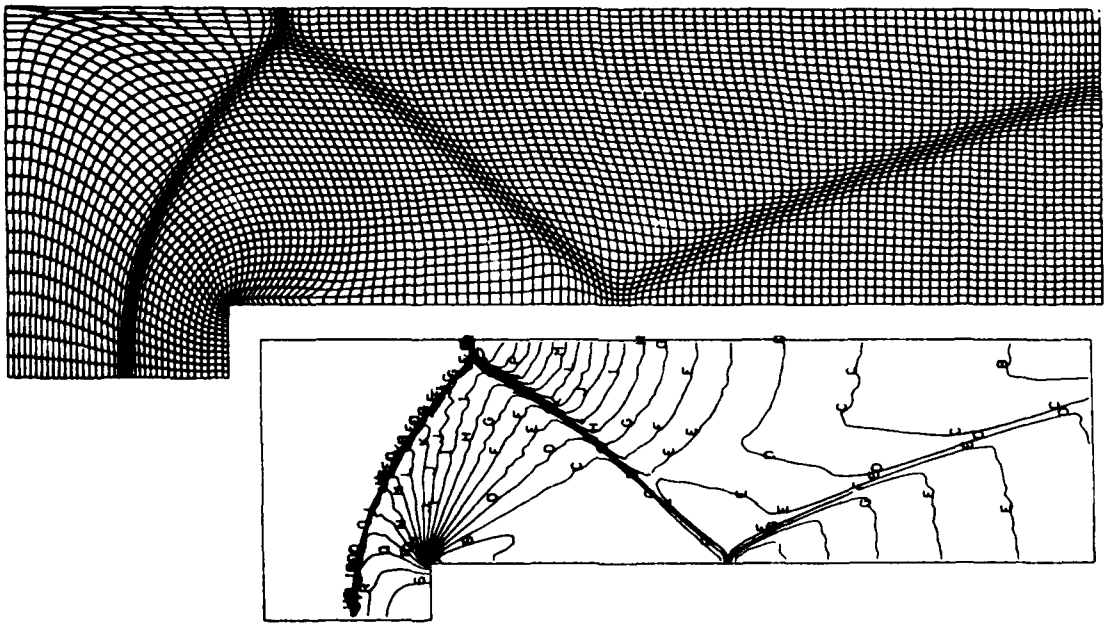


Figure 10. Shock reflection (from Brackbill 1982)

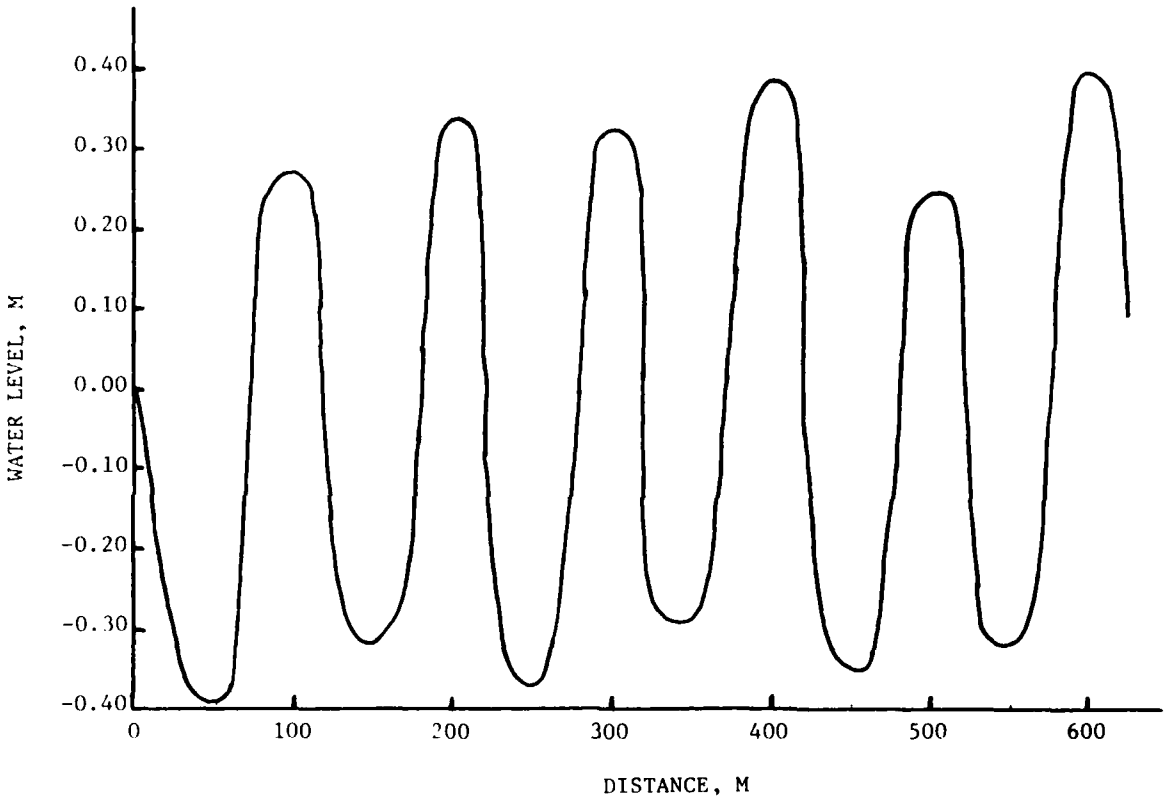


Figure 11. Propagation of short waves

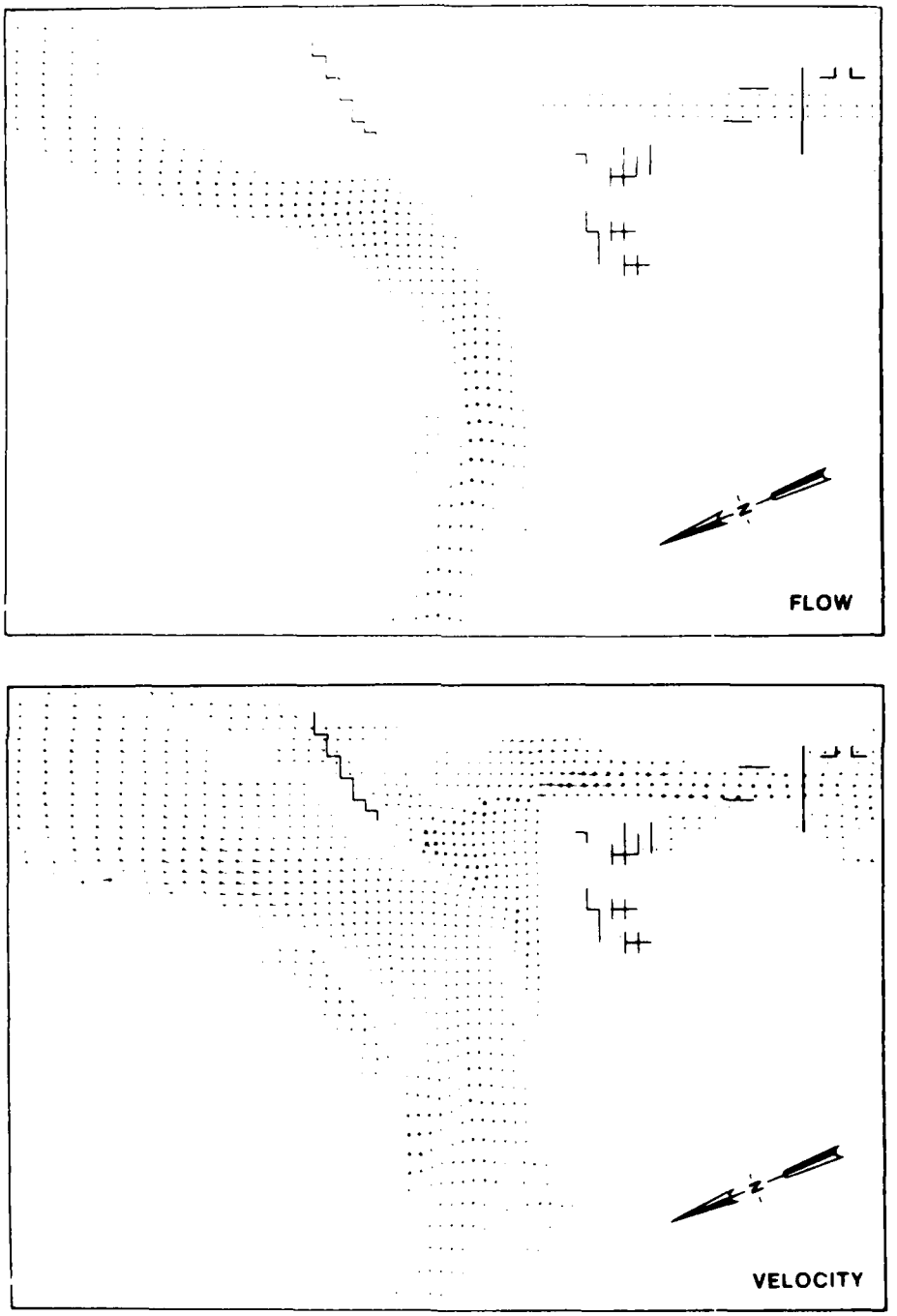


Figure 12. Current pattern in Coos Bay
(from Butler 1978)

early time

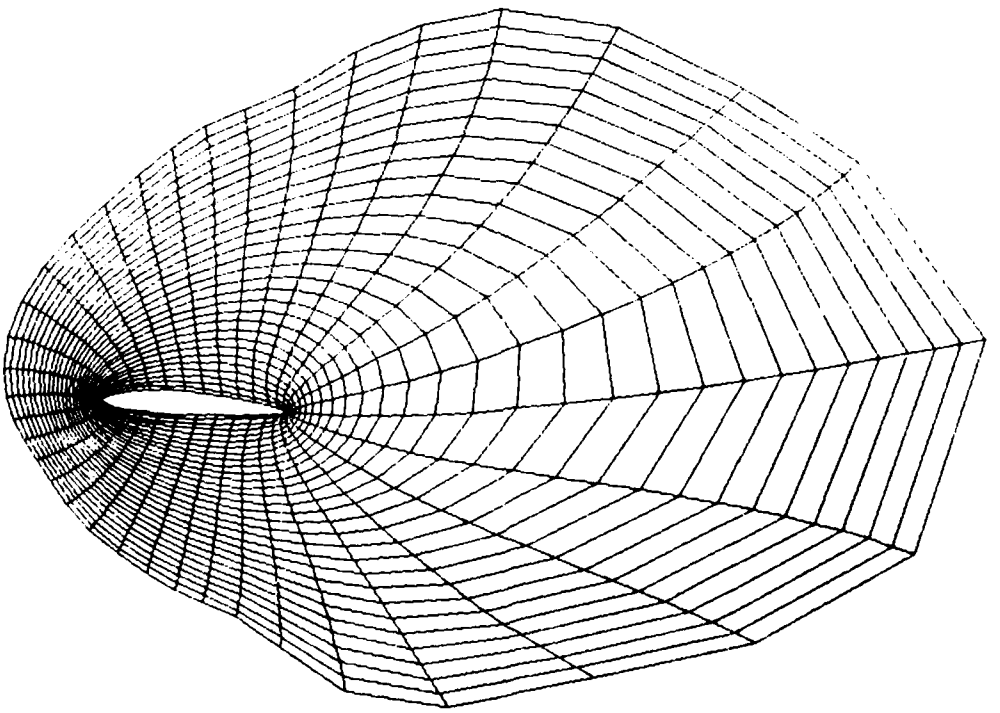


Figure 13. Expanding coordinate system
(from Reddy and Thompson 1977)

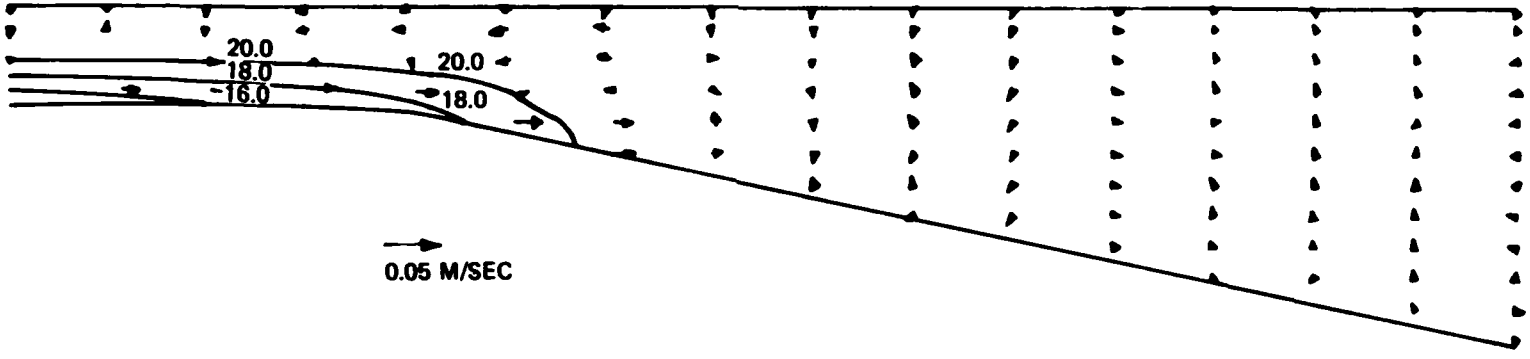
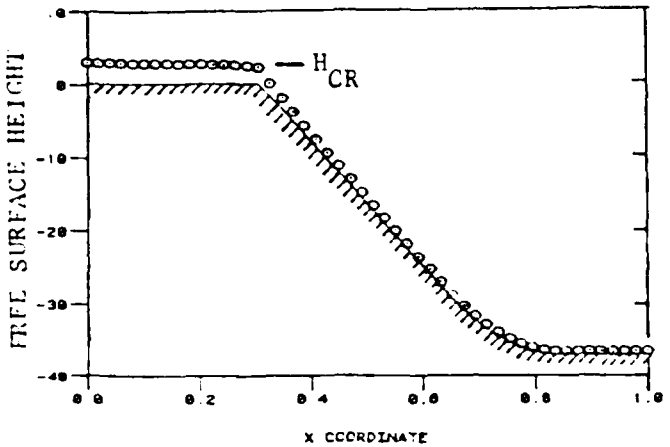
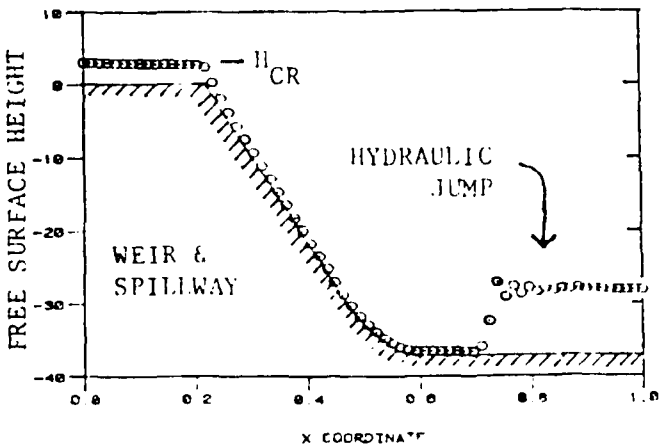


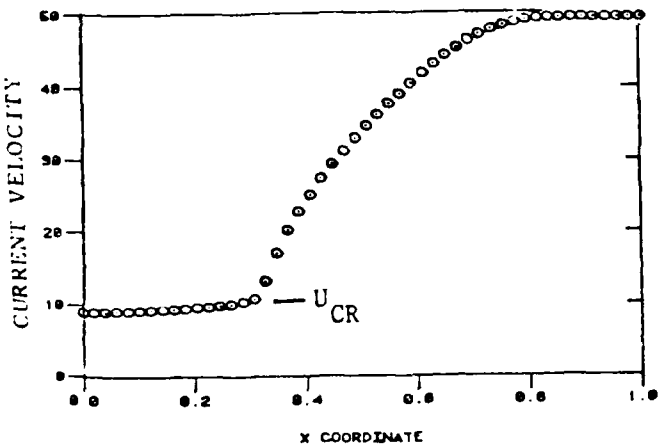
Figure 14. Velocities and isotherms at T = 10 min with cold-water inflow over lower half (from Johnson 1981)



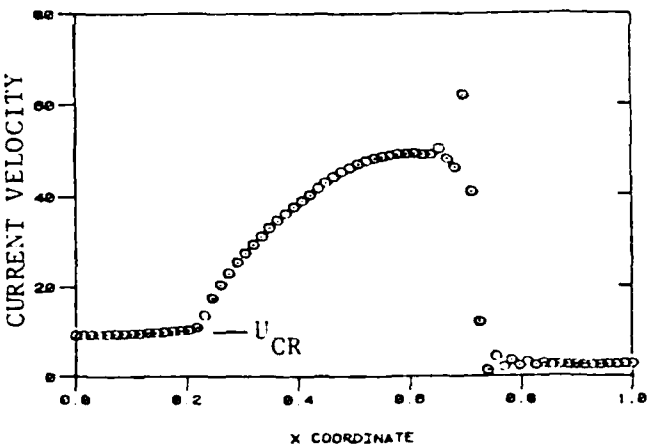
a) Free Surface, Without Hydraulic Jump



b) Free Surface With Hydraulic Jump



c) Velocity Distribution, No Hydraulic Jump



d) Velocity Distribution With Hydraulic Jump

Figure 15. Hydraulic jump computations (from Baker 1983)

APPENDIX A: HYDRODYNAMIC EQUATIONS AND APPROXIMATIONS

Approximations

1. The Navier-Stokes equations express the conservation of mass and momentum of a flow field and are the basic governing equations for the solution of any fluid dynamics problem. These equations written in tensor notation are

$$\text{Continuity:} \qquad \frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \tag{A1}$$

$$\text{Momentum:} \qquad \frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = \frac{-\partial P}{\partial x_i} + \rho g_i - 2\varepsilon_{ijk}\Omega_j \rho u_k + \frac{\partial T_{ij}}{\partial x_j} \tag{A2}$$

where

ρ = water density

t = time

u_l, u_j, u_k = tensor notation for velocity

x_i, x_j = tensor notation of spatial coordinate

P = pressure

g_i = acceleration due to gravity

ε_{ijk} = cyclic tensor

Ω_j = Coriolis parameter

T_{ij} = laminar stress tensor

In addition to the above equations, a conservation of energy equation must also be written for fluid flow problems with thermal effects. With the assumption of a constant specific heat and with the neglection of viscous dissipative effects, the energy equation can be written as the following transport equation for temperature T :

$$\text{Energy:} \qquad \frac{\partial T}{\partial t} + \frac{\partial (T u_i)}{\partial x_i} = \frac{\partial \left(D_{ij} \frac{\partial T}{\partial x_j} \right)}{\partial x_i} + \sum \text{sources} - \sum \text{sinks} \tag{A3}$$

where D_{ij} is the diffusivity coefficient. This equation states that the temperature can change as a result of advection by the flow field, the molecular

diffusion, and the actions of any sources and sinks of heat. As a matter of fact, this same equation applies to the transport of any constituent ϕ, where ϕ replaces T in the equation and appropriate sources, sinks, and boundary conditions are prescribed. For example, in the numerical modeling of the hydrodynamics of an estuary, a similar transport equation for the salinity would be required.

2. One additional equation remains to be written in order to close the system. An equation of state expressing the density as a function of the temperature and pressure (and salinity in estuarine modeling) must be employed:

$$\text{Equation of State:} \qquad \rho = \rho(T,P) \tag{A4}$$

3. With the closure of the system, six equations exist to be solved for the six unknowns--density ρ; three velocity components u, v, w; pressure P; and temperature T.

4. The above equations written with molecular values of viscosity and diffusivity are only applicable to laminar flows. Following Reynolds, the approach taken to make the equations applicable to turbulent flows is to assume that each dependent variable is composed of an average time-varying component plus a small randomly varying component about the average value. Integration over an increment of time produces a set of equations of the same form; however, these equations are now written with the time-averaged components as the dependent variables plus additional terms involving products of the turbulent fluctuations.

5. An important problem in numerical hydrodynamic models is the treatment of these turbulence terms. Most existing models utilize the concept of an eddy viscosity. Eddy viscosity models are relatively easy to use and can often give reasonable results if sufficient data are available to establish the validity of the model parameters. However, under flow conditions where the turbulence is generally not in equilibrium with the mean flow gradients, e.g., highly oscillatory flows; models that resolve the time rate of change, convection, diffusion, production, and dissipation of turbulence are required. These models are known as Reynolds stress models and involve the solution of dynamic equations for the second-order turbulent correlations along with the equations for the mean flow variables. With this type of model, a universal set of constants for a wide variety of flow conditions can be used.

6. Subject to the assumptions made in their derivation, the time-averaged equations are applicable to any turbulent fluid dynamics problem. An approximation usually made when applying the equations to hydrodynamic problems was first pointed out by Boussinesq. When variations of temperature are small ($\Delta t \leq \pm 10^{o}C$), variations in density will be less than one percent. These small variations can be ignored in general with one exception. The variability of density in the gravitaional term cannot be ignored. Hence, ρ is treated as a constant in all places except the body force term.

7. An assumption that is usually made in hydrodynamic models is that vertical accelerations are negligible when compared with the gravitational acceleration. Neglecting viscous terms also, the vertical momentum equation reduces to

$$\frac{\partial P}{\partial z} = \rho g \tag{A5}$$

where z is the vertical spatial coordinate and is positive downward. Equation A5 states that the pressure is hydrostatic.

8. When considering the coupling of the thermodynamics and the hydrodynamics of a water body, a distinction must be made between convective and "quasi-static" models. Convective models retain the complete vertical momentum equation and can simulate in full detail the convective overturning of unstable water masses, such as the upwelling of cells of warm water or perhaps the plunging of a cold-water inflow. Quasi-static models eliminate vertical accelerations due to buoyancy effects, which precludes the explicit treatment of free convection associated with unstable stratification. Convective overturning can only be handled as mixing along the vertical axis. Generally, models with the hydrostatic pressure assumption require far less computer time than the fully convective models.

9. A spatial-averaging technique similar to the turbulent time-averaging technique is used to yield either 1- or 2-D model (Ward and Espey 1971).* The spatial integration yields a set of equations with the time-averaged and spatially averaged components of the flow and thermodynamic variables as dependent variables plus terms involving products of the spatial fluctuations. As usually done with the turbulent fluctuations, eddy coefficients, referred to as

---

* References are at end of main text.

"eddy dispersion coefficients" to distinguish them from the turbulent eddy diffusion coefficients, are employed to close the governing equations.

10. Many depth-averaged models have been developed over the past 20 years or so; whereas width-averaged models have only been developed over the past 8 to 10 years. If the spatial integration is performed over the complete cross section, a 1-D model with variations allowed only in the longitudinal direction results. The governing sets of equations for both 1- and 2-D free-surface hydrodynamic models are given below.

One-Dimensional Cross-Sectionally Averaged

11. For the case of 1-D open-channel flows within rigid boundaries, the flow behavior can be adequately described by the Saint-Venant partial differential equations of unsteady flow. These equations, which are presented below, can be derived by integrating Equations A1 and A2 over a cross section.

Flow Continuity:

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = q_\ell \tag{A6}$$

Momentum:

$$\frac{\partial Q}{\partial t} + V \frac{\partial}{\partial x}(\beta Q) + \beta V \frac{\partial Q}{\partial x} - \beta V^2 B \frac{\partial y}{\partial x} + gA \frac{\partial y}{\partial x} = gA (S_x - S_f) + \beta V^2 A_x^y \tag{A7}$$

12. Equations A6 and A7 comprise the set of equations governing the motion of water in open channels in a 1-D sense and involve three unknowns, namely, the flow discharge Q, the flow depth y, and the frictional slope S_f. Other variables such as the lateral inflows and geometry data are expected to be known. To achieve closure of the system, an additional relation is required. This is provided by Manning's equation which relates the friction slope to the flow and channel characteristics

$$S_f = \frac{n^2 Q|Q|}{2.25 A^2 R^{4/3}} \tag{A8}$$

With this additional relation, one can then solve for the basic unknowns Q and y. Variables in the equations are defined as:

A = total cross-sectional area of channel

A_x^y = derivative of A with respect to channel distance at a constant flow depth

B = top width of water surface

g = acceleration due to gravity

K_e = coefficient in eddy head loss term

n = coefficient in Manning's equation

Q = flow discharge, in cfs

q_ℓ = lateral inflow rate of water

R = hydraulic radius

S_x = slope of channel bed

S_f = friction slope

t = time

V = average flow velocity

x = distance along the channel

y = depth of water in the channel

z = elevation of channel bed

β = momentum correction factor

$\partial/\partial t$ = derivative with respect to time

$\partial/\partial x$ = derivative with respect to channel distance

Two-Dimensional Vertically Averaged

13. An integration of the time-averaged 3-D equations, with T replaced by the salinity s, yields a set of equations with the time-averaged and depth-averaged components of the flow and salinity as dependent variables.

Continuity:

$$\frac{\partial \phi}{\partial t} + \frac{\partial (uh)}{\partial x} + \frac{\partial (vh)}{\partial y} = 0 \qquad \text{(A9)}$$

x-momentum:

$$\frac{\partial (hu)}{\partial t} + \frac{\partial (hu^2)}{\partial x} + \frac{\partial (huv)}{\partial y} = -\frac{h}{\rho}\frac{\partial P}{\partial x} + \frac{\partial \left(hD_{xx} \frac{\partial u}{\partial x} \right)}{\partial x} + \frac{\partial \left(hD_{xy} \frac{\partial u}{\partial y} \right)}{\partial y} + \tau_{s_x} - \tau_{B_x} + fhv \qquad \text{(A10)}$$

y-momentum:

$$\frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial(hv^2)}{\partial y} = -\frac{h}{\rho}\frac{\partial P}{\partial y}$$

$$+ \frac{\partial\left(hD_{yx}\frac{\partial v}{\partial x}\right)}{\partial x} + \frac{\partial\left(hD_{yy}\frac{\partial v}{\partial y}\right)}{\partial y}$$

$$+ \tau_{s_y} - \tau_{B_y} - fhu \qquad \text{(A11)}$$

Salinity:

$$\frac{\partial(hs)}{\partial t} + \frac{\partial hus}{\partial x} + \frac{\partial(hvs)}{\partial y} = \frac{\partial\left(hE_x\frac{\partial s}{\partial x}\right)}{\partial x} + \frac{\partial\left(hE_y\frac{\partial s}{\partial y}\right)}{\partial y} \qquad \text{(A12)}$$

An equation of state relating the water density to the salinity and water temperature (assumed constant) can be written as

$$\rho(s,T) = 1000\ \frac{P0}{AL} + ALO * P0 \qquad \text{(A13)}$$

where

$$AL = 1779.5 + 11.25T - 0.0745T^2 - (3.80 + 0.01T)s$$

$$ALO = 0.6980$$

$$P0 = 5890.0 + 38T - 0.375T^2 + 3s$$

In the above equations, the surface wind shear can be written as

$$\tau_{s_x} = \frac{W_c}{\rho_o}\rho_a v_w^2 \cos\alpha$$

$$\tau_{s_y} = \frac{W_c}{\rho_o}\rho_a v_w^2 \sin\alpha \qquad \text{(A14)}$$

and the bottom shear as

$$\tau_{B_x} = gu\sqrt{u^2 + v^2}\,/C^2$$

$$\tau_{B_y} = gv\sqrt{u^2 + v^2}\,/C^2 \qquad \text{(A15)}$$

The Coriolis parameter, f, is computed from

$$f = 2\omega_e \sin \lambda \tag{A16}$$

where

ω_e = earth's angular velocity

λ = angle of latitude of the center of the area being modeled

14. In order to finalize the above system of equations, it remains to couple the salinity computations with those of the flow field. Assuming that the pressure is hydrostatic, it can be shown that the horizontal pressure gradient terms can be written as

$$h \frac{\partial P}{\partial x} = h \frac{\partial P_a}{\partial x} + h\rho g \frac{\partial \phi}{\partial x} + \frac{1}{2} h^2 g \frac{\partial \rho}{\partial x}$$

and (A17)

$$h \frac{\partial P}{\partial y} = h \frac{\partial P_a}{\partial y} + h\rho g \frac{\partial \phi}{\partial y} + \frac{1}{2} h^2 g \frac{\partial \rho}{\partial x}$$

Substituting Equations A14-A17 into Equations A9-A11 would then yield the final form of the governing equations. Variables in the equations are defined below:

C = Chezy coefficient

D_{xx}, D_{yy} = diagonal components of eddy viscosity tensor

D_{xy}, D_{yx} = off diagonal components of eddy viscosity tensor

E_x, E_y = components of eddy dispersion tensor

g = acceleration due to gravity

h = water depth

P = pressure

P_a = atmospheric pressure

s = salinity

T = temperature

u, v = components of velocity

vw = wind speed

W_c = resistance coefficient

x, y = Cartesian coordinates

α = wind direction

λ = latitude of center of modeled area

ρ = water density

ρ_a = density of air

ρ_o = reference water density

τ_{B_x}, τ_{B_y} = components of bottom shear stress

τ_{s_x}, τ_{s_y} = components of bottom shear stress

ϕ = water-surface elevation

ω_e = earth's angular velocity

$\partial/\partial t$ = time derivative

$\frac{\partial}{\partial x}, \frac{\partial}{\partial y}$ = space derivatives

Two-Dimensional Laterally Averaged

15. The equations below governing flow in relatively narrow and deep bodies of water are obtained by integrating the 3-D equations over the width. Since both salinity and temperature effects may influence the vertical stratification, a transport-diffusion equation for both salt and temperature is given.

Longitudinal (x-direction momentum)

$$\frac{\partial}{\partial t}(UBh) + \frac{\partial}{\partial x}(U^2Bh) + (u_b w_b b)_{k+1/2} - (u_b w_b b)_{k-1/2}$$

$$+ \frac{1}{\rho}\frac{\partial}{\partial x}(PBh) - A_x \frac{\partial^2}{\partial x^2}(UBh) + (\tau_z b)_{k+1/2} - (\tau_z b)_{k-1/2} = 0 \qquad \text{(A18)}$$

with

$$t_z = C^*\rho_a/\rho W_a^2 \cos\phi \quad \text{(surface)}$$

$$= A_z \, \partial U/\partial z \quad \text{(interlayer)}$$

$$= gU|U|/C^2 \quad \text{(bottom)}$$

Internal continuity

$$(w_b b)_{k-1/2} = (w_b b)_{k+1/2} + \frac{\partial}{\partial x}(UBh) - qBh \tag{A19}$$

Total depth continuity

$$\frac{\partial(\xi b)}{\partial t} - \sum_k \frac{\partial}{\partial x}(UBh) = \sum_k qBh \tag{A20}$$

Vertical (z-direction) momentum

$$\frac{\partial P}{\partial z} = \rho g \tag{A21}$$

Heat balance

$$\frac{\partial}{\partial t}(BhT) + \frac{\partial}{\partial x}(UBhT) + (w_b bT)_{k+1/2} - (w_b bT)_{k-1/2}$$

$$- \frac{\partial}{\partial x}\left(D_x \frac{\partial BhT}{\partial x}\right) - \left(D_z \frac{\partial BT}{\partial z}\right)_{k+1/2} + \left(D_z \frac{\partial BT}{\partial z}\right)_{k-1/2} = \frac{H_n Bh}{V} \tag{A22}$$

Salinity balance

$$\frac{\partial}{\partial t}(Bhs) + \frac{\partial}{\partial x}(UBhs) + \left(w_b bs\right)_{k+1/2} - \left(w_b bs\right)_{k-1/2}$$

$$- \frac{\partial}{\partial x}\left(D_x \frac{\partial Bhs}{\partial x}\right) - \left(D_z \frac{\partial Bs}{\partial z}\right)_{k+1/2} + \left(D_z \frac{\partial Bs}{\partial z}\right)_{k-1/2} = sqBh \tag{A23}$$

Equation of state

$$\rho = [1000\ P_o/(LA + 0.698\ P_o)] \tag{A24}$$

where

$$P_o = 5890 + 38T - 0.375T^2 + 3s$$

$$LA = 1779.5 + 11.25T - 0.0745T^2 - (3.8 + 0.01T)s$$

Variables in the equations above are defined below:

A_x = x-direction momentum dispersion coefficient (m^2/s)

A_z = z-direction momentum dispersion coefficient (m^2/s)

b = estuary or river width (m)

B = laterally averaged width integrated over h (m)

C = Chezy resistance coefficient, $m^{1/2}/s$

C^* = resistance coefficient associated with wind

D_x = x-direction temperature and salinity dispersion coefficient (m^2/s)

D_z = z-direction temperature and salinity dispersion coefficient (m^2/s)

g = acceleration due to gravity (m/s^2)

h = horizontal layer thickness (m)

H_n = source strength for heat balance ($C^o m^3 s^{-1}$)

k = integer layer number, positive downward

P = pressure

q = tributary inflow or withdrawal (m^3/s)

s = laterally averaged salinity integrated over h (ppt)

t = time (s)

T = laterally averaged temperature integrated over h (C^o)

U = x-direction, laterally averaged velocity integrated over h (m/s)

u_b = x-direction, laterally averaged velocity (m/s)

V = cell volume ($B \cdot h \cdot \Delta x$) (m^3)

W_a = wind speed (m/s)

w_b = z-direction, laterally averaged velocity (m/s)

x, z = Cartesian coordinates: x is along the estuary center line at the water surface, positive to the right and z is positive downward from the x-axis (m)

ζ = surface elevation (m)

ρ = density (kg/m^3)

ρ_a = air density (kg/m^3)

τ_z = z-direction shear stress

ϕ = wind direction (rad)

16. The concept of eddy coefficients is utilized to represent the effect

of both time averaging and spatial averaging of the equations. The horizontal coefficients, A_x and D_x, are assumed to be constant; whereas the vertical coefficients, A_z and D_z, are dependent upon the stratification as reflected by the local Richardson number, R_i, i.e.,

$$A_z = A_{z_o}\left(1 + 3.33R_i\right)^{-3/2}$$

$$D_z = D_{z_o}\left(1 + 10R_i\right)^{-1/2} \tag{A25}$$

where

$$R_i = \frac{\frac{g}{\rho}\left(\frac{\partial \rho}{\partial z}\right)}{\left(\frac{\partial U}{\partial z}\right)^2} \tag{A26}$$

and A_{z_o} and D_{z_o} are the vertical coefficients for no stratification. Due to the hydrostatic pressure assumption, unstable stratification cannot be modeled in a convective fashion and thus is handled in a diffusive manner by increasing D_z to its stability limit.

17. The laterally averaged horizontal pressure gradient in the longitudinal momentum equation contains the density driving force. Using the expression for the hydrostatic pressure, the horizontal pressure gradient can be divided into its two components of the barotropic (surface slope) gradient and the baroclinic (density) gradient to yield:

$$\frac{\partial p}{\partial x} = -g\,\frac{\partial \xi}{\partial x} + g\int_{\xi}^{z}\left(\frac{\partial \rho}{\partial x}\right) dz \tag{A27}$$

Substituting Equations A25-A27 into Equations A18 and A22-A23 would then yield the final form of the governing equations for the computation of laterally averaged stratified flows.

END

FILMED

5-85

DTIC